

开源时代

OPEN SOURCE TIMES

2010年4月刊 总第十九期

开源业界:

开源世界里谁的影响力最大?
Novell继续加大中国战略投资



技术新知:

Linux内核学习的方法论
RHEL5.4下LAMP平台搭建及测



社区扫描:

微软调整业务战略 不再敌视开源社区
Ubuntu 10.04测试版 加入社交功能

行业观察:

Linux是否已经迎来了自己盈利的春天
发展Linux: 中日两国之比较

本期推荐:

2010数据库技术大会侧记

2010年4月2-3日, 正值清明节, 由IT168(ChinaUnix、ITPUB、IXPUB)主办的2010数据库技术大会胜利落幕。这是一次代表着当前国内数据库技术最高水平的大会, 是国内各行各业DBA一次面对面的精彩交流机会。会议的两天的议程精彩纷呈, 主流数据库、国产数据库和开源数据库悉数登场。



内容目录

刊首语.....	4
开源业界.....	5
IBM 发布 Power7 难阻 Unix 市场颓势.....	5
Novell Linux 守得云开 与微软协议关系不大.....	6
朝鲜自主 Linux 操作系统：红星 Red Star.....	7
IDC：服务器 OS 逆转 Windows 击败 Linux.....	13
进军企业服务器 Ubuntu 准备好了吗？	14
Novell 继续加大中国战略投资.....	16
品牌推广 Linux 基金会开设在线商店.....	17
开源世界里谁的影响力最大？	18
日企推新技术令 Android 系统 1 秒钟启动.....	19
前 MySQL CEO 呼吁更多创业公司加入开源运动.....	21
微软推新驱动程序支持在 Hyper-v 上运行 Linux.....	22
法开源商业软件公司指控 IBM 垄断大型机市场.....	23
ARM 看好 Android 在中国数字电视领域前景.....	24
开源硬件能否复制开源软件的成功？	25
社区扫描.....	30
Google 赠送手机给 Android 开发者.....	30
GNOME 3：面向“Task”的桌面.....	30
Digg 用 NoSQL 替代 MySQL.....	31
Mozilla 复查其开源协议 或与 Apache 协议兼容.....	32
微软调整业务战略 不再敌视开源社区.....	32
Eric Raymond：几种计算机语言的评价.....	32
Nvidia 放弃对开源驱动的支持.....	35
Ubuntu 10.04 测试版 加入社交功能.....	35
行业观察.....	36
Linux 是否已经迎来了自己盈利的春天.....	36
发展 Linux：中日两国之比较.....	37
打折出售：Linux 厂商 Novell 将花落谁家？	38
开源软件未来发展趋势：更少利润更多理念.....	39
收购之后 MySQL 前路多艰.....	40
开源硬件能否像开源软件一样流行.....	41
Linux 操作系统在中国市场的四大发展障碍.....	43
联想的软肋是开源软件？	45
技术沙龙.....	48
2010 数据库技术大会精彩回眸.....	48
技术新知.....	51
Linux 内核学习的方法论.....	51

Apache 服务器文件的详细配置.....	66
centos 5.4+Layer7 封 qq、msn 和 p2p 下载.....	70
Linux 上 Postgres Plus 数据库备份和恢复.....	74
alsa 音频库的移植	83
RHEL5.4 下 LAMP 平台搭建及基本测试.....	87
网友热评.....	105

本期编辑：周荣茂 覃里

美工：冯瑶

编校：王文冰

投稿邮箱：rmzhou@staff.chinaunix.net

本刊网址：<http://linux.chinaunix.net/ebook/>

刊首语

北京的春天也太春寒料峭了，眼看都快 5 月了，还是这么冷。

4 月初的数据库技术大会已经结束，谢谢大家的关注和参与！由于酒店预定的原因，我们被迫将会议时间定在了清明节期间，开始，有不少人跟我建议，担心因为节日影响参会人群的积极性。说实在的，对于搞 DBA 的人来说，节日消减不了他们的积极性。

事实上，本次大会实际数据库相关参与者达到了 550 人，加上庞大的出版社、赞助商等参与者，大会总人数接近 700 人。其中，本次大会，数据库专家（演讲专家、CTO、首席 DAB、知名数据库图书作者等）超过了 150 位，而其他参会者也是数据库领域的资深骨干工程师。

大会的外地参会人数比例接近 1/3。甚至，还有远道来自我国台湾省的几位技术朋友。4.2 号的晚宴真应该搞起来，这样，就可以请来自我国台湾的朋友们，讲讲那边的技术工作环境。

数据库大会结束，我们也将在全国各地展开更具体的技术会议和沙龙活动，这次我们将选择第一天的商业会议和第二天的开源技术交流相结合的方式。目前主要选定的地区有，上海/杭州/南京、广州/深圳，西安、成都和重庆，其他地方还在考虑之中，如果有《开源时代》读者想和我们合作，请联系我们！

关于这次开源技术交流，我们想和第一天的商业会议想区别，会场会选择学校或者培训机构，主要面向的听众是高年级学生、研究生和刚刚工作的技术人员，目的是更广泛地宣传自由软件和开源软件，为整个中国的自由软件和开源技术社区打下一个坚实的基础，技术交流的形式我们也将安排多种多样，比如技术讲座、InstallFest、开源企业参观交流等，如果你有什么好的形式和建议，也欢迎和我们联系！

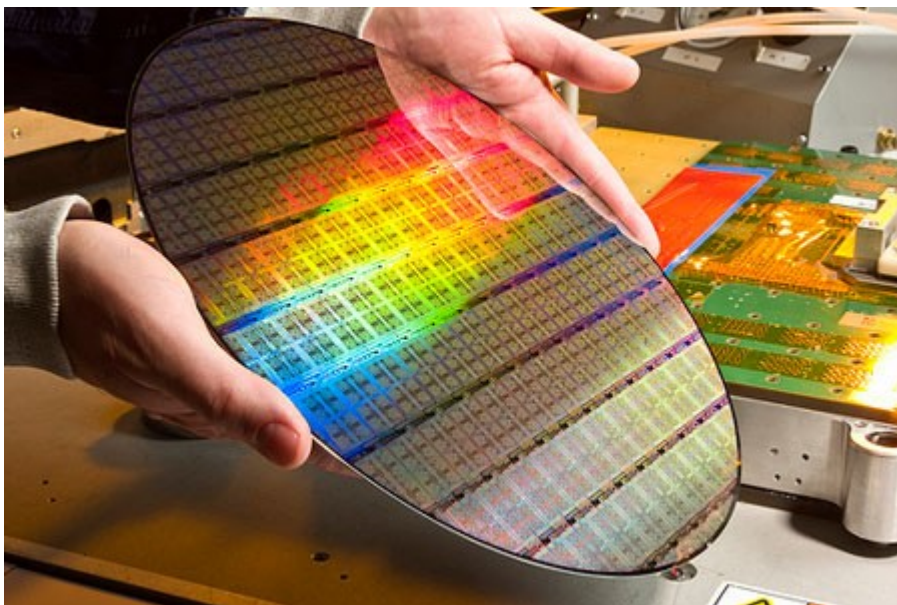
----ChinaUnix 社区编辑：周荣茂

开源业界

IBM 发布 Power7 难阻 Unix 市场颓势

IBM 在北京盛大发布三年磨一剑的 POWER7 处理器，与上一代 POWER6 相比，POWER7 处理器采用 45 纳米工艺制造，由双核直接升至 8 核，加之每个核心可实现 4 线程，这样最多可以支持 32 线程，性能上相较 POWER6 提升达 4 倍。

在发布会上，IBM 罕见的将 POWER7 与英特尔至强处理器做比较，这在以往是不可能的事情，这不仅体现了 IBM 对 POWER7 信心满满，也意味着，POWER7 所处的 UNIX 服务器阵营已经感受到以英特尔至强为代表的 X86 服务器带来的压力。



据 IDC 数据显示，去年第四季度 Unix 服务器收入下降 18.1% 至 39 亿美元，而 x86 服务器销售额增长 12.6% 至 73 亿美元。

作为 UNIX 服务器市场的领头羊，IBM 寄望 POWER7 的发布能够提振 UNIX 服务器市场，将基于 POWER7 的 POWER 服务器从处理金融、电力等行业的关键业务拓展至实时分析等新兴行业。只是市场会如 IBM 所愿吗？

随着虚拟化、开发平台转移、IT 整合与简化、提高性能和可用性、降低软硬件的总体拥有成本等需求的不断增强，Unix 向基于 X86 平台 Linux 迁移已经成为不可逆转的趋势。

以银行业为例，当前，大多数中小规模商业银行面临着产品创新、业务可持续发展与核心业务系统技术条件限制的矛盾。由于商业银行的 IT 系统是在不同时期开发的，缺少统一的 IT 规划、技术规范和技术框架，造成了系统之间的集成比较困难。传统的 Unix 服务器存在成本高、技术壁垒多等问题，不利于银行 IT 系统朝着开放、简单、标准化的方向发展。

许多新兴的区域性商业银行在选择基础架构时，Unix 平台在某些情况下可能还是用户的首选，但 x86 架构和 Linux 平台在开放性、稳定性、安全性等方面表现出的优势给商业银行用户提供了一种

新选择。

东亚银行(中国)有限公司首席信息官林丽认为，相对于 Unix 服务器，X86 架构的刀片服务器不仅价格便宜，而且容易扩展和整合，更适合应用所需。东亚银行的数据库系统仍旧运行在 Unix 平台上。不过，在对业务进行拆分后，大量的业务应用被迁移到 Linux 平台上，保留在 Unix 平台上的业务量不是太大，因此也没有必要配备大型的 Unix 服务器。

“我比较熟悉小型机，但是现在已经变成了 Linux 的赞同派。以前，小型机意味着可靠、稳定、管理简单；现在，x86 架构及 Linux 系统在技术上已经达到小型机的水平。刀片+Linux 的解决方案更高效、更灵活。”林丽打比方说，“银行的核心应用、账务系统可以放在主机或小型机上，而前端应用可以放在开放平台上。”

从封闭到开放，商业银行走上 Unix 向 Linux 的迁移之路并不是刻意为之。现有系统架构的不灵活、过度配置、投资大、管理成本高等问题给业务的发展造成了瓶颈，这才促使商业银行客户居安思危，希望通过 IT 架构的调整和优化，尽量降低成本，避免进入一个恶性循环。

Novell Linux 守得云开 与微软协议关系不大

Novell 坚持了 7 年的 Linux 业务终于实现了收支平衡，他们良好的完成了一年前的承诺。

在上周 Novell 发布的截至 1 月 31 日的 2010 财年首份季度收入报告中，公司的营业额为 2 亿 200 万美元，比起 2009 年同期的 2 亿 1500 万略有下滑。而在净收入方面的情况好了许多，Novell 报告说公司的净收入为 2000 万美元或每股 0.06 美元，比起去年同期的 1100 万美元或每股 0.03 美元有了很大幅度的增加。

而在重点提到的 Linux 业务上，Novell 首席财务官 Dana Russell 报告说，Linux 平台的产品收入增长了 6%，达到 3700 万美元，并且取得了收支平衡。



2003 年，Novell 以 2100 万美元收购了 SUSE Linux 开始进入 Linux 业务。一直到 2009 年 5 月，Russell 透露说 Novell 的 Linux 的业务依旧无法赚钱，但他当时已经敢于宣称将在未来 12 至 18 个月的时间取得收支平衡。

现在结果出来了，Russell 为投资者们提前带来了好消息。

“我们原先所做的承诺是让 Linux 在今年年底收支平衡，”他说，“尽管在收入报告中我们没有详细的分析 Linux，但它确实达到了收支平衡，这是一个重要的里程碑。”

与微软面临续展

三年前，Novell 与微软签订了关于 Linux 支持服务的合作协议，这在当时反响极大。如今，这份协议的三年首期已经结束，因此关于微软在 Linux 业务上所起的效果究竟如何，Novell 在上周也多少透露了一些。

然而结论并不乐观：Novell 称 Linux 业务取得收支平衡的主要原因是行业的强劲增长，而并非依靠与微软的合作协议。

事实上，Novell 从微软的合作协议中获得的收入反而在逐年明显的减少。在 2006 年两家企业签订协议时，微软付给 Novell 2 亿 5000 万美元，收购了 SUSE Linux 的支持许可，用于销售给微软和 Novell 的共同的客户。在 07 年的 Novell 财报上，这项合作使 Novell 获益 3.6 亿美元。现在，微软和 Novell 正面临着更新这批服务许可证，但并不会是曾经的价格。

“微软的 Linux 支持许可证将在本季度更新，事实上我们这三年从与微软的合作中获得的收入比原先协议中预计的低了不少，当然这也是意料之中的，” Russell 说，“因为其他 Linux 支持业务显著增长，因此抵消了一部分与微软合作的收入下降。”

Russell 表示收入下降的原因是 Novell 是按照折扣价格向微软销售 Linux 支持许可，而微软也以折扣卖给企业。

“原先的许可证我们是从微软得到售价的 45%，”他说。“而大多数这些许可证都以相当大的折扣卖给了那些大型的战略客户，特别是在这一项目的早期。”

比如，全球最大的零售商沃尔玛就是最早就与微软和 Novell 签署协议的大客户之一，这当然是一家需要给它折扣的企业。而从总数上说，Novell 和微软在 2009 年底曾表示他们的共同客户已经超过了 400 家。

“因此我们对续约的期望很大，我们希望能够得到接近售价的 80%至 90%，特别是那些大型的战略客户，” Russell 说，“只要计算一下，你就会明白这能为我们带来什么，在财务上会有非常大的帮助，这是我们希望这个项目带给我们的。”

朝鲜自主 Linux 操作系统：红星 Red Star

据国外资讯网站 RT 报道，朝鲜（North Korea）不仅拥有自主的、与基于美国的万维网相互独立的区域网络，而且还拥有自行研发的红星 Red Star 操作系统。据悉，红星 OS 是去年由朝鲜 IT 技术人员在 Linux 操作系统的基础上改进而来的。俄罗斯学者 Mikhail 利用他在金日成大学交流之际获得了这款红星 OS。他花费了 5 美元在距离宿舍五分钟步行路程的地方购得两张安装光盘。系统共分为客户端和服务端两种版本，第三方软件可以另外花钱购买。

（编者注：有来自 CU 网友的爆料，出于以前对麒麟的关注习惯，我看到这个新闻的时候，一下就想到了，Red star linux.这个名字我很熟悉，我甚至在麒麟的 qq 群中专门讨论过这个事情。如果我没有记错的话，red star linux 应该是北京麒麟威方公司的产品！！现在外面没有这个宣传了，百度应该还能搜出来。另一方面，北朝鲜根本不具备这种能力的 linux 组装水平。所以该版本完全是银河麒麟援外版在朝鲜的推广活动。或者是麒麟威方公司将版权完全转让给北朝鲜。）

废话不多说，下面就让我们一睹为快：

1、拆包后的安装盘，上下共两张；



2、启动 Logo 画面，红星 OS 支持硬盘或网络下载安装，整个过程约为 10-25 分钟。启动安装程序后，用户会发现整个过程与 Windows 十分相似，但更加快速、简单，而且只有朝鲜语一种语言版本；



3、登录界面：默认为 Root 用户名，右侧的 Lim 是游客账号；

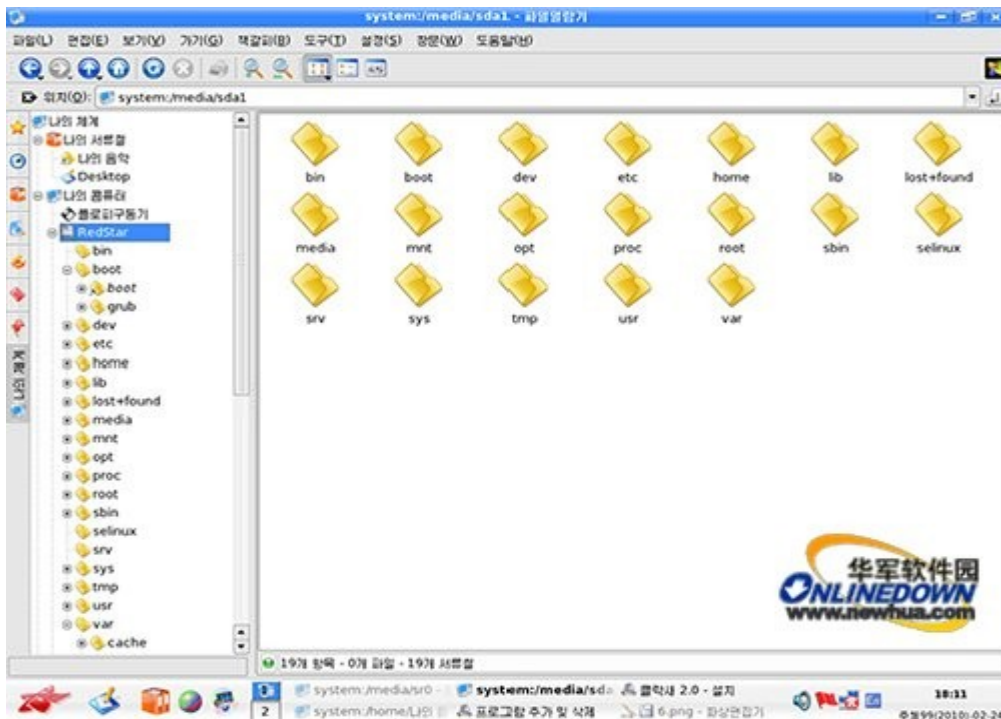


俄罗斯学者 Mikhail 利用他在金日成大学交流之际获得了这款红星 OS。他花费了 5 美元在距离宿舍五分钟步行路程的地方购得两张安装光盘。系统共分为客户端和服务端两种版本，第三方软件可以另外花钱购买。

4、Red Star 桌面环境



5、文件布局



与纯正的 Linux 相比，红星 OS 算不上最轻便的操作系统，它的最低硬件需求为奔 III 800MHz 处理器、256M 内存、3G 固态硬盘容量。

桌面环境布局与同类产品十分相似，提供“我的电脑”、“回收站”等图标。界面左侧下方则是开始菜单，旁边是快速启动按钮

6、应用程序一览



系统集成应用程序十分有限，诸如“My Country”浏览器（更换外壳的 Firefox）、简单的 Word 字处理软件、PDF 文档阅读器、视频播放软件、虚拟磁盘管理软件。

7、小游戏（象棋）



红星 OS 采用独特的键盘标准，分别可用于输入英语、朝鲜语、汉语和俄语；娱乐应用中包含四款益智小游戏，分别是扫雷、中国象棋、碎石机、花图纸牌；

8、其他应用程序



光盘的其他内容有与 Openoffice 很类似的“we”办公组件，“Pigeon”电子邮件服务、传真通

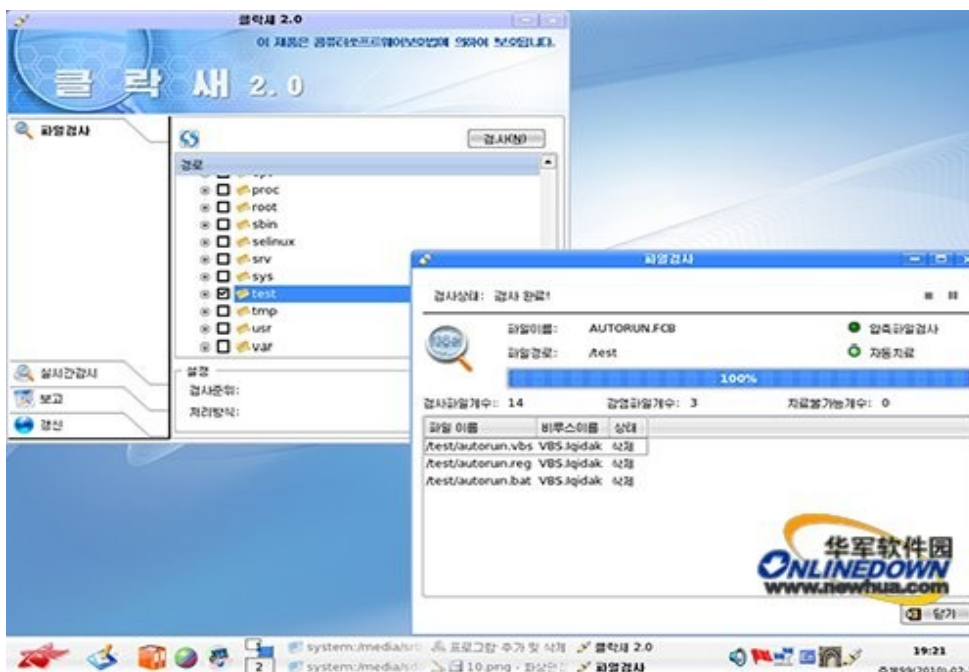
信服务、“Pyongyang Fortress” 防火墙和计算器/日历等；

9、又见 Firefox 的身影



浏览器中默认的搜索引擎是 Google，但无法搜索到任何有用的信息，因为朝鲜的网络是完全封闭的。

10、安全软件



红星 OS 整合的 Antivirus 2.0 套装（朝鲜文，不认识）算是唯一货真价实的正品，并没有改写某款开源代码，全部代码都是自行编写，具备一定的查杀恶意程序能力。另外，Mikhail 使用运行 Warcraft 3 等大型应用软件时，系统表现平稳；

11、“开始”功能菜单



IDC：服务器 OS 逆转 Windows 击败 Linux

一些 Linux 爱好者认为尽管 Windows 几乎垄断了桌面领域，但是在服务器市场 Linux 更有优势。但是 IDC 的新的调查表明目前这种状况出现了改变，在最近至少一年的时间里，Windows 在服务器市场击败了 Linux，市场份额比超过三比一。

IDC 近日发布了全球服务器市场的报告，主要集中在服务器硬件，IBM 领先全球服务器市场，HP、DELL、Sun 和富士通依次居后。

IDC 统计数字：2009 年第四季度服务器市场（硬件）份额和营收情况

Vendor	Q4 2009 Revenue (Millions)	Market Share	Q4 2008 Revenue	Market Share	Revenue Growth Q409/Q408
1. IBM	\$4,590	35.4%	\$4,910	36.4%	-6.5%

2. HP	\$3,950	30.5%	\$3,920	29.1%	0.8%
3. Dell	\$1,489	11.5%	\$1,425	10.6%	4.5%
4. Sun	\$1,032	8.0%	\$1,247	9.3%	-17.3%
5. Fujitsu	\$595	4.6%	\$556	4.1%	7.2%
Others	\$1,296	10.0%	\$1,422	10.5%	-8.9%
All Vendors	\$12,952	100%	\$13,480	100%	-3.9%

硬件方面并不出人意料，而有趣的是关于操作系统市场份额的数字。这份研究还表明，在 2009 年第四季度，Windows Server 占据的市场份额为 73.9%，Linux 为 21.2%，比值超过了三比一。这些数字和一年前几乎是一样的，仅仅回落了 0.1 个百分点。

在营收方面，Windows 在 2009 年第四季度营收 54 亿美金，而 Linux 营收仅为 19 亿美金。从这些数字可以看到，在服务器操作系统市场 Windows 无论是在份额还是营收上都远远超过了 Linux，这意味着 Windows 不仅垄断了桌面，在服务器市场也表现出明显优势，并且没有减缓增长的迹象。

进军企业服务器 Ubuntu 准备好了吗？

从各方面来看，Red Hat 都是企业 Linux 领域无可争议的领导者，但 Ubuntu 正在证明它也能够应付这一挑战。凭借着桌面端的良好声誉和强大影响力，Ubuntu 正在一步步的进军商业服务器，与 Red Hat 展开强力竞争，相信很快企业评估 Ubuntu 的时刻即将到来。

和你们中的许多人一样，我原先对企业 Linux 市场的认识也一直是 Red Hat Enterprise Linux (RHEL) 是无疑的主导者，而 Novell 的 Suse 在稍远的地方占据第二名的位置。这种想法一直持续到去年五月，我阅读了 Eclipse Survey 2009 的调查结果并且发现了两个非常有趣的关于 Linux 的统计数据，这让我不得不重新考虑这个观点。

我很惊奇地发现在调查中有接近 15% 的 Eclipse 开发者回答说他们正在开发环境中使用 Ubuntu，我本想将这归功于 Ubuntu 的用户体验比起 Fedora/RHEL 或 OpenSuse/Suse 来说要好一些，因为用户界面毕竟是 Canonical 下了一番大功夫的（在 2009 年的亚特兰大 Linux 大会上，Canonical 强调了 Ubuntu 的 9 大特点）。但后来我意识到，就受访者的回答来看，Ubuntu 在服务器市场的表现也一样好。Fedora/RHEL 虽然依旧领导着市场，但也只是微弱领先 Ubuntu 一个百分点而已。尽管从收入和出货量的角度来看，根据 IDC 的估计，Red Hat 依然至少占有着 60% 以上的 Linux 的市场份额。

关于 Ubuntu 采纳的惊人真相

自那以后，我就一直迫使自己的眼睛注意着 Ubuntu 在企业领域的成长。本周早些时候，我发现著名的电影数字效果工作室 Weta Digital——他们的作品包括“阿凡达”、“第九区”、“心灵传输者”还有“指环王”——是大规模使用 Ubuntu 的（相关阅读：制作《阿凡达》的超级数据中心）。Dustin Kirkland，来自 Canonical 的一位 Ubuntu Server 的核心开发者在 Linux.conf.au 2010 社区的谈话中写下了以下内容：

“他们有一个了不起的数据中心，能够为如今的电影大片制作需要的特殊效果。数据中心有一个 2Pb 的磁盘阵列，10Gbps 的网络和 35000 核心（超过 4000 多台惠普的刀片服务器），即使这样，仍然需要 48 小时来渲染他们的图像序列。”



“Ubuntu 是这一切的核心，运行所有的渲染节点，还有 Weta Digital 90% 的桌面。” Kirkland 指出他的农场（他称之为“渲染墙”）实际上就是一个 Ubuntu Server 农场，而并非是来自 Red Hat 的产品，这和一些媒体报道所说的不一样。

Weta Digital 的数据中心相当惊人，尤其是在绿色数据中心的实践方面。不难看出，Ubuntu 为 Weta Digital 所运行的工作完全可以被划分为企业级。

“每天有 1 万多项工作，估计有 130 万到 140 万任务要完成。这些每秒 24 帧的电影每一帧都包含着导演与演员的努力，要花费许多小时渲染出来。”

Canonical 的策略是在客户端桌面继续强势发展 Ubuntu，这个领域 Red Hat 已经不能插足。然而从 Eclipse 的调查以及 Weta Digital 的使用效果看来，作为服务器操作系统，Ubuntu 也已经不容忽视。另一方面，在云计算的进程上，Ubuntu 也与红帽步调一致，一同开始了其面向客户的云服务。

软件厂商们应该开始认证 Ubuntu 了

当你考虑在你的企业中采纳 Ubuntu 尤其是作为服务器操作系统时，毫无疑问你会有充分的理由关注它对企业应用的支持如何。在现今的情况下 RHEL 肯定比 Ubuntu 拥有大得多的软件开发者环境和机制，这意味着比起 Ubuntu 来，企业应用在 RHEL 中更容易进行测试和认证。

但是，随着 Ubuntu 的成长和 Novell Suse 的不确定因素，面向企业的软件供应商将不会再耽搁 Ubuntu 的认证工作了，这会帮助 Ubuntu 成功快速的走入企业领域。

Novell 继续加大中国战略投资

近日，全球著名数据中心安全、管理和开源技术提供商 Novell 公司宣布了对华投资的进一步计划，主要涉及增加更多的研发人员、销售人员和工程师以及市场推广，在中国增设新的区域分支机构与扩大本地团队，加深与本地合作伙伴关系等内容，目标在 2012 年(3 年内)将中国业务收入提高两倍。该计划是 Novell 继 2005 年第一批投资计划后的又一大规模再投资，表明了 Novell 对中国市场的无比重视，同时也标志着 Novell 将更好地推动中国 Linux 的发展和利用，进一步积极开展全面的区域扩展攻势，努力建立和完善中国各重点区域的本地化支持，为中国用户提供更加安全有效的解决方案。



早在 2005 年，Novell 董事会就已决定在中国战略投资。在过去的 4 年里，Novell 中国业绩连连攀升，一直保持两位数的增长，营业额已提高 11 倍，2009 年在金融危机情况下收入甚至增长了 150%。同时，Novell 成功地在北京、上海、广州、深圳建立办事处，赢得了众多包括电信行业、金融服务业和政府部门在内的关键性大客户，及关键的 ISV/IHV。值得注意的是，在过去 4 年里，Novell 已成功转型为提供最工程化、操作互动性最强的 Linux 平台，以及一系列集成的 IT 管理软件、虚拟化及云计算的公司，通过其基础架构软件和紧密的合作伙伴关系，Novell 可以完美地将混合式 IT 环境集成一体，让人与技术协作融洽。

“Linux 已经从建立起数据中心到桌面的主流技术，而中国 Linux 的发展将很大程度上影响和决定全球 Linux 的发展速度。” Novell CEO Ron Hovsepian 先生表示：“在过去的 4 年中，Novell 在中国的业务表现强劲，我对中国市场也给予了厚望。扩大对中国的投资正显示了 Novell 公司对中国市场的信心与承诺，同时我们也希望中国市场为 Novell 公司创造更多的机会。”

最新的投资计划主要涉及以下几方面内容：

增加 Novell 在中国销售方面的投入，建立更多的分支机构，销售覆盖更多省份。Novell 期望以此进一步提升对中国市场和用户的支持，建立良好、顺畅的基础设施和服务系统，充分发挥当地分支机构的便利和优势，为他们提供更完善、更专业的本地化服务，提高 Novell 的服务水平。

进一步加强渠道的建设。到 2012 年，Novell 渠道将扩展至 10 个二级城市，每个城市都将发展核心经销商，与地方合作伙伴和地区代理合作，销售 Novell 产品。并加深与独立软件开发商(ISV)的合作。

加大研发的人员投入。中国研发中心的地位已逐渐在全球研发队伍中突显，其研发任务包括开发 Linux 桌面技术与产品、进行 Linux 国际化与本地化的研究，以及从事高端技术的研发，特别是云计算产品方面。从全球来看，Novell 将继续加大对 Linux 系统、身份识别、安全管理、云计算方面的研发投资，现在的研发投入已经占到了 Novell 公司收入的 19%，未来还将继续增加。这些措施在未来五到十年都会对公司产生非常重大的影响。在北京、上海、广州、深圳、厦门建立 5 个 Novell 解决方案展示体验中心(Novell Solution Demo Center)。这将大大方便用户的使用，为其提供最便捷的技术支持和服务，让用户能够最亲近、最快速地了解 Novell 产品，为广大用户带来最高价值的服务。

开展更多市场活动，包括在多个城市举办研讨会，技术培训，渠道会议等等宣传介绍 Novell 及其解决方案，并与一些大客户合作建立虚拟化数据中心，在大学开展培训项目等等。

“继续加大对中国市场的投入，表明了 Novell 将把更多国际最先进的 Linux 产品、技术以及管理经验带给中国的用户和合作伙伴，与中国的合作伙伴共同推进中国 Linux 的发展和中國本地软件行业的发展。” Novell 东亚区总裁张先民博士说。“Novell 中国已为跨入一个新的时代做好了准备，也为大力推动中国的 Linux 发展做好了准备。”

品牌推广 Linux 基金会开设在线商店

在苹果 iTunes 在线商店模式获得成功之后，很多厂家纷纷效仿发布自家在线商店。如今 Linux 基金会也宣布上线了最新的在线商店 store.linux.com，新的在线商店上销售各类与 Linux 相关的产品，主要包括 T 恤衫、杯子、帽子之类的小物品。

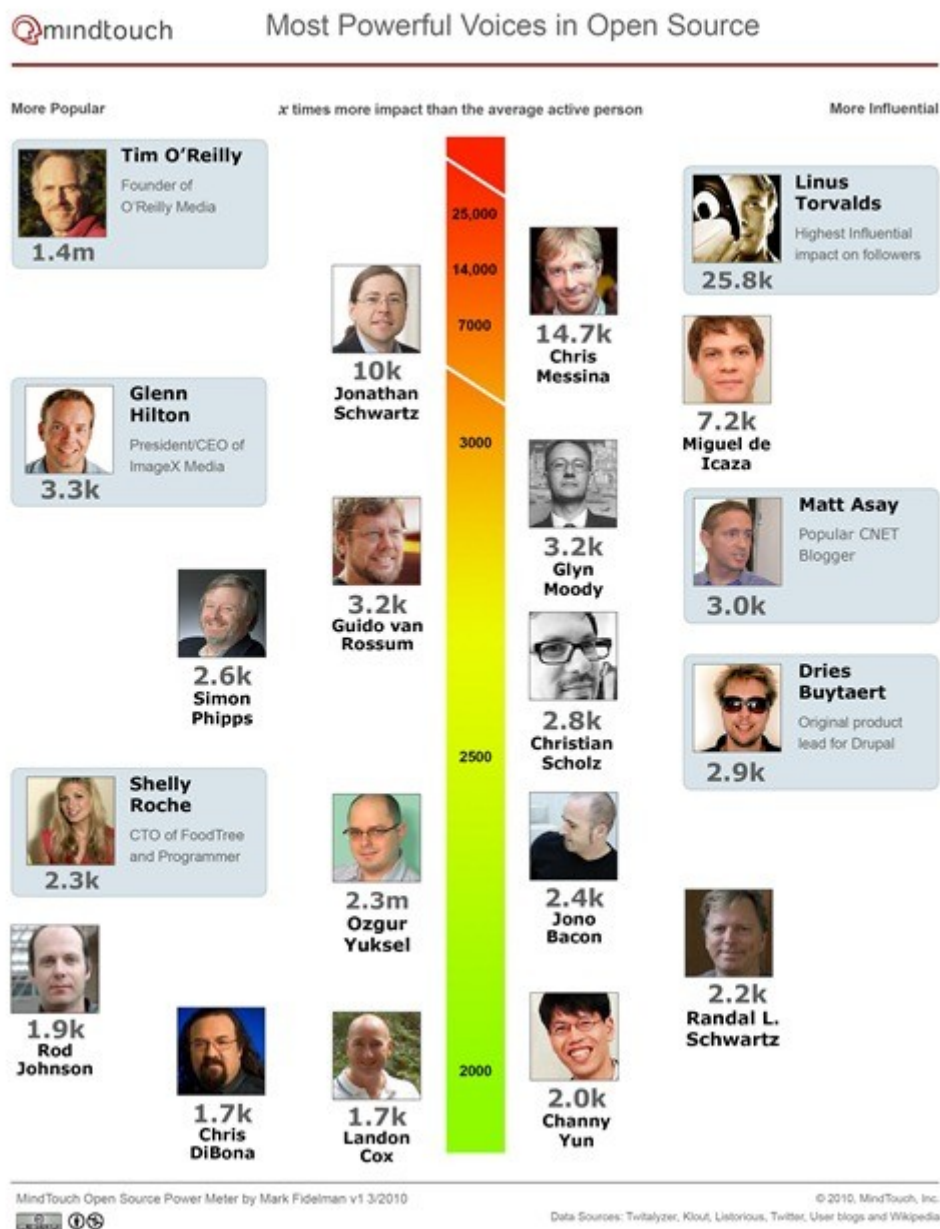


Linux 基金会通信主管 Jennifer Closer 表示：“带有‘Let's Have Code Sex’、‘FSK the Establishment’、‘Fork You’之类短语的商品是这个新的在线商店的特色，由 Linux 基金会独家提供。” Linux 基金会同时还宣布了 Linux.com T 恤设计大赛，社区成员可以提交自己设计的以 Linux 为主题的 T 恤设计，活动截至 4 月 11 日。届时 Linux 基金会将评出前五名优胜参赛者，并在网站上对其作品展开公开投票。投票截止 6 月 5 日，最终获胜者将获得免费采集 8 月份举办的 2010 年波士顿 LinuxCon 大会的机会，其设计作品将在 Linux.com 商店上进行销售。

根据 Linux 基金会的说法，该商店的所有收入将用于 Linux 开发、活动等。

开源世界里谁的影响力最大？

开源协作公司 MindTouch 3 月 17 日在今年开源商业会议上发布了他们根据 Google Trends 和各种 Twitter 分析工具得出的最具影响力开源意见领袖榜。



其中，著名技术出版公司 O'Reilly 的创始人 Tim O'Reilly 因其 Twitter 的 143 万关注者而荣登榜首。Linux 之父 Linus Torvalds 虽然很少上 Twitter，博客更新也不勤，而且多是写自己私人活动，仍然以无可比拟的强大实力而名列第二。其他前 10 名意见领袖还包括：Google 的开源宣传员 Chris Messina，Sun 的前 CEO Jonathan Schwartz，Mono 创始人 Miguel de Icaza，知名 Drupal 设计和开发公司 ImageX Media 的创始人 Glenn Hilton，技术作家、开源图书《Rebel Code》的作者 Glyn Moody，Ubuntu 开发公司 Canonical 的 COO Matt Asay，Drupal 之父 Dries Buytaert 和 Python 之父 Guido van Rossum。

前 50 名榜单中，比较知名的还包括：最近离开 Oracle 的 Sun 公司首席开源官 Simon Phipps，MySQL 前 CEO Marten Mikos，Ubuntu 社区经理 Jono Bacon，美女程序员和企业家、知名技术播客 Shelly Roche，Perl 大牛 Randal Schwartz，Spring 之父 Rod Johnson，Mono 前身 Ximian 联合创始人 Nat Friedman，Google 开源和公共部门工程经理 Chris Dibona 等等。

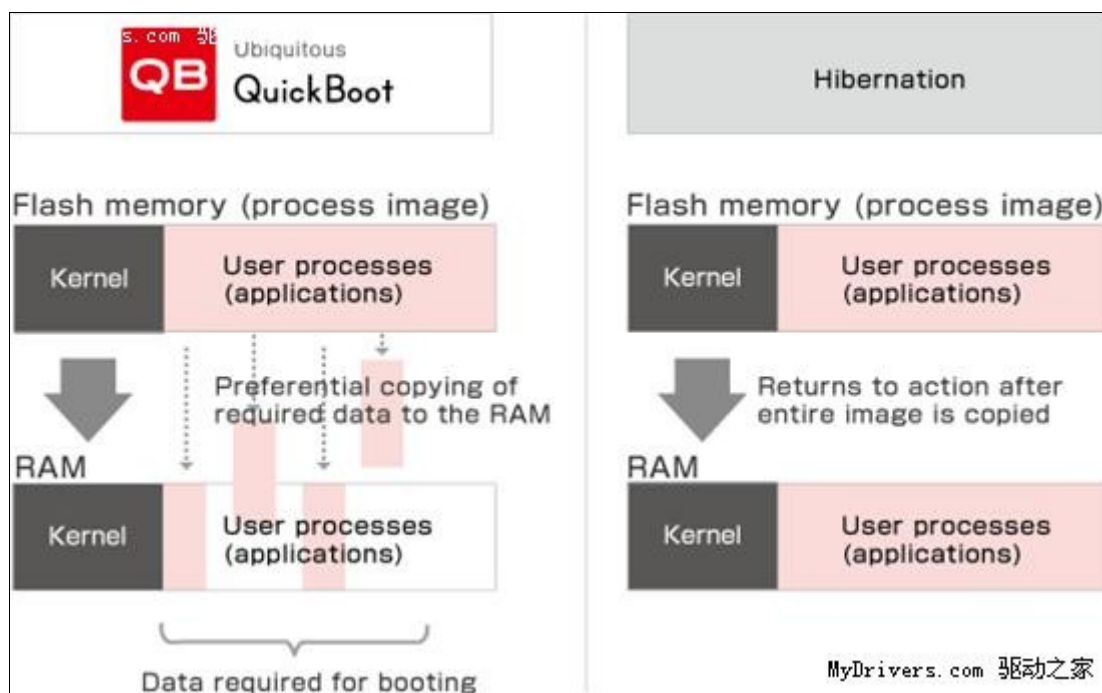
日企推新技术令 Android 系统 1 秒钟启动

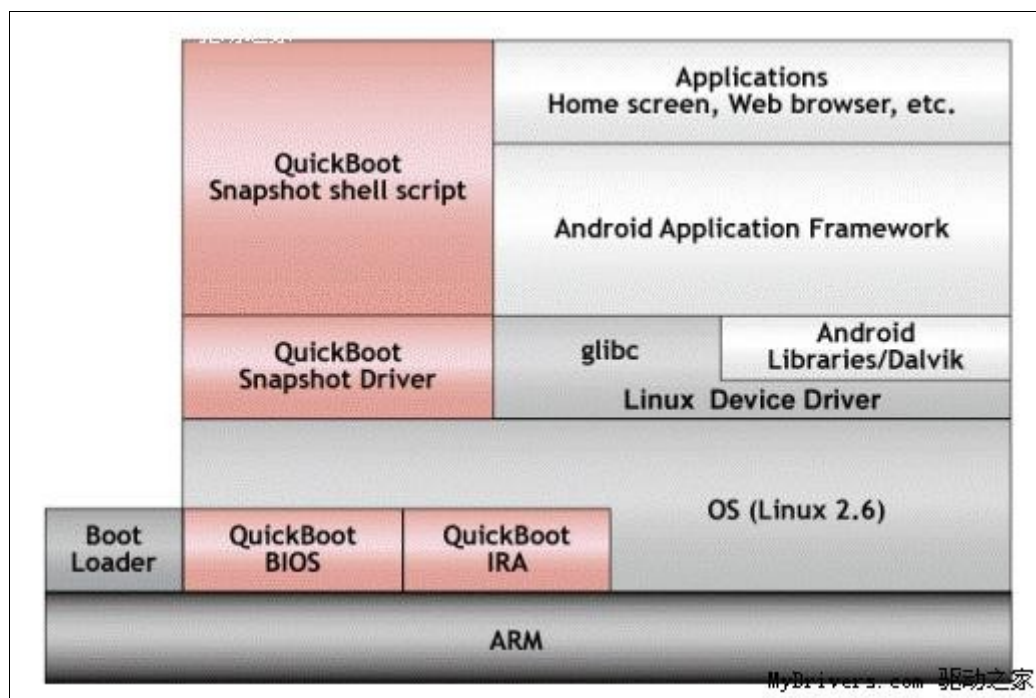
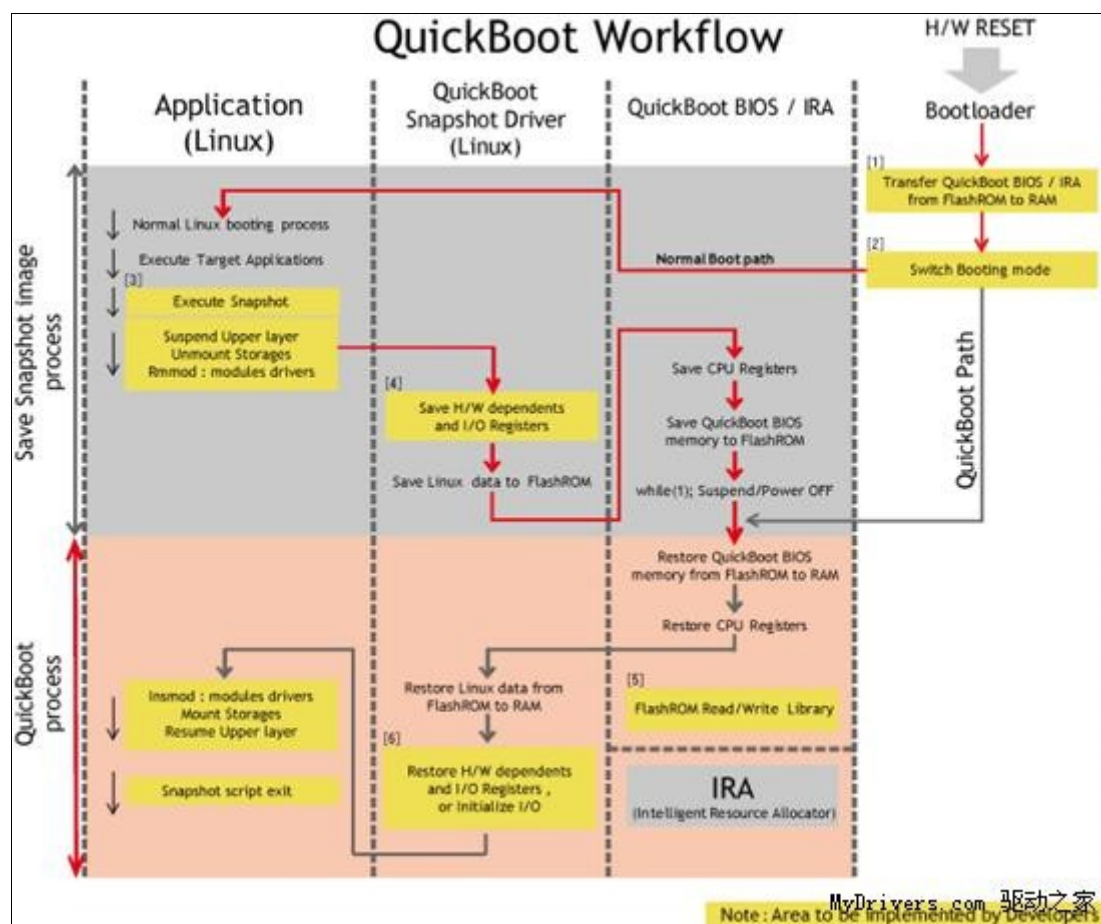
日本嵌入式软件开发商 Ubiquitous 公司日前宣布，他们已经开发出了一种名为 QuickBoot 的技术，可以令电视、机顶盒、汽车电脑、智能本、智能手机等 ARM 架构平台产品在 1 秒钟内完成 Linux 或 Android 系统的冷启动过程。

该技术的核心在于，将系统启动必需的核心数据优先从非易失性存储设备（闪存）复制到内存，完成初步启动工作后再依次传输其他应用程序数据。相比常见的休眠模式需要将所有内容复制到内存后才完成启动，这种 QuickBoot 模式的速度更快，而且启动速度也与系统内存容量无关。

在 Ubiquitous 的演示中，两合同样由 Atmark 公司制造的 Armadillo-500 FX Android 开发平台，均搭载 Freescale i.MX31 400MHz ARM11 处理器，128MB SDRAM。基于 QuickBoot 技术的系统在 1.4 秒内即完成了 Android 系统启动，可以正常进行操作，对比系统则花了将近 20 秒。

Ubiquitous 表示，QuickBoot 技术目前支持 ARM9、ARM11 和 Cortex-A8 系列平台，Linux 和 Android 系统，他们将向客户厂商提供一套 QuickBoot SDK R1.0 以实现该技术。而未来，他们计划将该技术推广到 Cortex-A9 系列的多核平台，以及更多的操作系统上。本周在北京举行的中国国际广播电视信息网络展览会上，Ubiquitous 就将展示 QuickBoot 技术。





前 MySQL CEO 呼吁更多创业公司加入开源运动

据国外媒体报道，MySQL 前 CEO 马顿·米克斯（Marten Mickos）周三在 EclipseCon 2010 会议上强调，开源已不再是失败者，但需要有更多创业公司通过开源软件赚钱。

软件业专家在 EclipseCon 公司举办的会议上讨论了开源的未来。米克斯强调，通过开源赚大把钞票的创业公司太少。倒是拥有成熟商业模式的 IBM、微软、甲骨文和 Apache、Eclipse、Linux 和 Mozilla 基金会等公司为开源发展不懈努力。

米克斯说：“我们希望有非常棒的创业公司通过开源赚上数百万、甚至数十亿美元，但目前还没有实现这一目标。开源领域有大量成功的赚钱公司如红帽、Jboss、MySQL 及 XenSource，但光有这些还远远不够。”



米克斯最近出任开源云技术创业公司 Eucalyptus 的 CEO。MySQL 于 2008 年被 Sun 收购，甲骨文今年 1 月完成对 Sun 的收购后，获得 MySQL。米克斯说：“MySQL 能够为开源运动贡献大量 GPL 代码，因为该领域一直有营收。”米克斯称，有部分人士批评 MySQL 的双重许可计划，即公司一方面通过开源提供软件，同时向要求提供支持服务的用户收费。

Apache Software 基金会总裁贾斯汀·伊伦科朗茨（Justin Erenkrantz）也强调有必要保证开源营收。伊伦科朗茨说：“需要维持商业公司赚钱的生态系统，便于开发人员为开源做贡献的同时能获得一份收入，这就需要开源社区和创业公司共同保证开源的未来。”

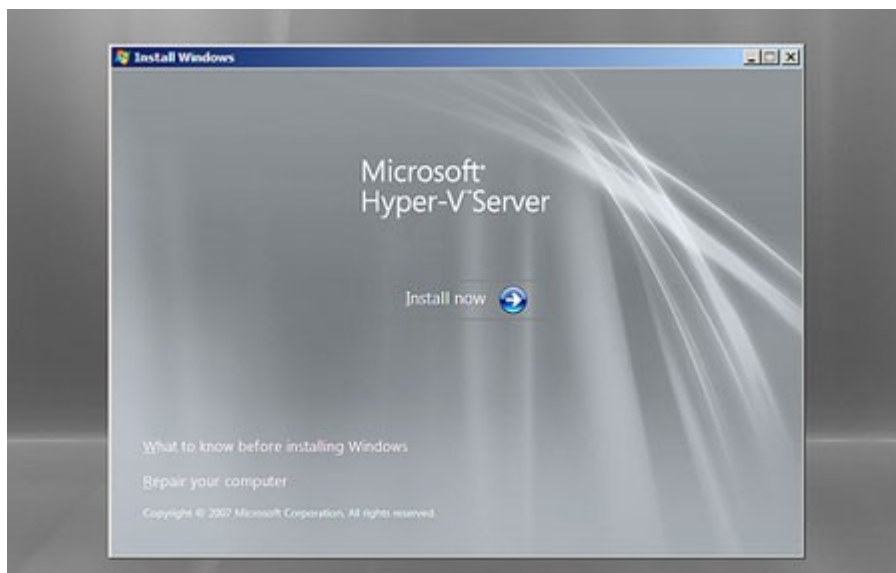
开源分析与咨询公司 RedMonk 分析师斯蒂芬·奥格雷迪（Stephen O'Grady）对开源未来表达了不同的观点，强调分散化的开发和基础设施。奥格雷迪还强调通过对信息进行聚合、分析，挖掘数据潜力。

尽管有部分观点认定开源无法创新，只是使市场商品化，但奥格雷迪指出，开源已在云技术领域引发了创新。他同时承认，开源也是市场商品化的积极推进者。

微软推新驱动程序支持在 Hyper-v 上运行 Linux

微软发布了一套驱动程序，可以使红帽公司不同版本的 Linux 产品在 Hyper-V 上运行。这些驱动程序目前仍存在一些限制，但它们得到了微软和红帽都将提供支持的完全的保证。

具体地说，这些驱动程序支持 RHEL 5.2、5.3 和 5.4。这些驱动程序还支持 SUSE Linux Enterprise Server 10 SP2 x86 和 x64 以及 SUSE Linux Enterprise Server 11 x86 和 x64，不过这些都是老新闻了。



微软是在去年 7 月首次高调向 Linux 内核提交上述驱动程序的(出于显而易见的原因，这是它到目前为止第一次，也是惟一的一次，对 Linux 的贡献)。这些驱动程序已经经过了与红帽，当然还有 SUSE，配合工作的测试。去年 10 月，红帽和微软宣布说他们将把各自的虚拟化合作伙伴计划联合起来，验证了它们的产品可在对方的虚拟机上运行。那么，是什么造成微软那么长时间才向公众发布这些红帽驱动程序呢？

这是笔者向微软的 Windows 虚拟化团队主要程序经理 Vijay Tewari 提出的问题之一。简而言之，延期是由一个问题造成的：两家公司没有就如果红帽操作系统作为 Hyper-V 客户机运行的话由谁来支持这个操作系统达成协议。这个支持问题几周前得到了解决，因此驱动程序公开发行。

Linux 在 Hyper-v 上运行仍存在限制

首先，让我们来指出这些限制。运行在 Hyper-V 上的红帽虚拟机局限于一个虚拟处理器，而这将限制 Linux 工作负载的规模。不过，Tewari 说，这种局限并非红帽所特有的。SUSE 也局限于一个虚拟处理器。虚拟处理器的数量对于不同版本的 Windows Server 也是不同的。WS 2008 R2 VM 可使用 4 个虚拟处理器，WS3003 为 3 个。Tewari 说，微软正在为增加运行在 Hyper-V 上的 Linux 服务器的虚拟处理器数量而努力。

其次，这些组件本身不提供对鼠标的支持，虽然 Citrix 开发了解决这个问题的一种添加件：Project Satori。Tewari 说，如果你直接使用服务器，鼠标可以工作。但是，没有这个添加件，如果以远程桌面连接方式接入服务器并通过远程连接启动客户操作系统的控制台，鼠标就不可用。

这些驱动程序是与 2009 年 7 月提交给 Linux 内核的那些驱动程序同时诞生的，但是随着这些最

初的产品经过来自 Linux 社区的补丁和更改的改动，此次驱动程序下载与 Linux 内核这间的代码库分道扬镳。微软说，今后几个月后，它将再把它们集成到一个代码库中。但是，Tewari 说，由于 Hyper-V 驱动程序提交给了当时的稳定版内核 2.6.32(版本 2.6.33 上个月底推出)，这种分道扬镳是不可避免的。并不是所有的商用版本都使用这一版内核。例如，这些驱动程序支持基于 2.6.18 和 2.6.27 内核的红帽。

作为重新统一代码工作的一部分，微软承诺增加一些改进，如解决鼠标问题的修改。

法开源商业软件公司指控 IBM 垄断大型机市场

据国外媒体报道，法国开源软件商业解决方案公司 TurboHercules 周二向欧盟委员会竞争总署正式提起诉讼，指控 IBM 阻止其用户使用 Hercules 软件。



Hercules 是一款在欧洲开发的开源大型机模拟器，在过去 10 年中，该模拟器一直由全球的志愿者社区来维护。TurboHercules 成立于 2009 年，主要是基于 Hercules 开源项目向用户提供商业解决方案。该公司指控 IBM 阻止用户在更为便宜的非大型机上的不同操作系统中运行用户应用程序。

Hercules 开源项目早期开发人员、TurboHercules 董事长罗格·鲍勒(Roger Bowler)表示：

“Hercules 是一项创新性开源技术，可以惠及许多大型机用户。但 IBM 却正采用将大型机操作系统与其硬件捆绑的方式，阻止用户使用 Hercules，这种做法让 TurboHercules 无法为那些希望采用开

源解决方案的大型机用户提供服务。”

鲍勒称：“TurboHercules 并不是要与 IBM 作对，我们开始曾请求 IBM 在公平合理的前提下向用户提供大型机操作系统授权，以方便用户在部分特定情况下使用 Hercules。但 IBM 不仅拒绝了这一请求，更在 10 年之后突然发难，声称 Hercules 开源模拟器侵犯其知识产权。我们这才认识到，作为一家小公司，我们唯一的希望是向欧盟委员会提起诉讼。”

ARM 看好 Android 在中国数字电视领域前景

CCBN2010 是三网融合政策出台之后的第一个大型展会，三网融合以及广电 NGB 建设的巨大市场不但吸引了国内厂商，更吸引许多国外厂商带来了专门为中国的三网融合市场、NGB 市场设计的产品、技术以及解决方案。ARM 联合其合作伙伴北京时代飞腾、韩国 Telechips 公司、台湾 Skymedia、日本 Ubiquitous 等公司展示了基于 ARM 架构的 Android 平台的互联网电视、网络播放器等解决方案。

ARM 看好 Android 在中国数字电视领域前景

据悉，CCBN 期间，从芯片公司到系统厂商甚至运营商都对 Android 表示了异乎寻常的兴趣。而从去年到今年，国外到国内很多厂商都转到基于 ARM 架构的 Android 软件平台，ARM 期望其在家庭应用上能得到比较好的应用和发展。据了解，目前 Android 开源唯一支持 ARM 架构。

对于三网融合政策带来的机遇多还是挑战多，ARM 中国总经理兼销售副总裁吴雄昂表示，“三网融合政策不仅是对 ARM，对中国的整个电子产业都是非常好的一件事情”，他认为电视、机顶盒等消费电子产品都会因为网络化而更新换代，从而给产业带来机遇。ARM 家庭应用市场经理邹诚认为：NGB、三网融合使硬件、软件、应用、网络融合加速，会要求更加开放的平台，而目前的开放平台中 Android 在数字电视领域是最受欢迎的。

对于 Android 转向家庭市场，ARM 认为电视与手机分辨率不同、视频不同、操作不同，如果能在家庭成功的话，主要是因为免费的、开源的、跨平台的，软件技术做的比较好能带来好的软件应用。吴雄昂认为，目前已经有很多公司采用了 ARM 架构，因此能与中国的三网融合、NGB 很快接轨，而他也相信 ARM 的开放模式能帮中国产业扫除障碍。

把最大价值留给合作伙伴

据介绍，ARM 的商业模式是开放合作，通过第三方提供服务和软件，合作方式主要是技术功能开发。吴雄昂反复强调，ARM 是一家提供 IP 的公司，成功的方案都是其合作伙伴做的。“ARM 商业模式最成功之处是把最大价值留给合作伙伴”，他这样说。据他介绍，目前，ARM 在全世界有 650 多家合作伙伴，在国内有 60 多家。

ARM 的合作伙伴也都对 Android 前景表示了强烈信心，他们认为 Android 给电视厂家提供了全新平台，传统电视与 Android 结合给电视厂商更多商业模式，可以做更多自己的应用。Skymedia 系统工程部处长杨景尧认为，2008 年国产电视出现销售倒挂，目前基本扳回局面，互联网电视功不可没。而第一代互联网电视炒功能，新一代产品需要服务来驱动，Android 给其很好的平台。日本 Ubiquitous 公司现场展示了其 Android 系统 1 秒启动的快捷。

吴雄昂很看好开放平台在中国的发展前景，据他透露，ARM 将会加强在软件这块的部署，今年下半年会更紧密团结合作伙伴，实质推动 Android 在电视上的标准化。另外，他也直言不讳的表示，国外 Android 支持声音没有中国大，但法国电信等对其表示了极大关注，如果有成型方案完全有可能得到应用。

对于流媒体网记者关注的中国市场的投入，吴雄昂透露，ARM 在中国持续增加人力支持，而在 2008 年经济不好的时候中国也是唯一保持经济增长的区域。他表示，今年 ARM 在中国市场的主要任务是帮芯片伙伴增加出货量。为此，ARM 一方面加大售后支持团队，另一方面今年要在中国成立针对应用软件的研发中心。

开源硬件能否复制开源软件的成功？

开源软件的成功是过去数十年来最伟大的成功故事之一。Apache HTTP Server 是全球最流行的 Web 服务器软件，Linux 在与 Unix 和其他专利操作系统的竞争中巩固了自己的地位，而 Mozilla 的火狐浏览器这些年来已成为微软 IE 的强劲对手。

那么同样的哲学——免费并公开传播底层源代码和开发规范，与众多对程序设计做过贡献的开发者分享——能否适用于硬件的设计呢？是否会有那么一天，我们大家都能用上按照开源模式设计的智能手机、上网本或者其他器具呢？是否有可能我们喜欢开源硬件会胜过像 iPhone 这样的专利产品呢？

毕竟，今天在 PC 上用 CAD 和原理图设计软件来设计一个设备——包括其电子部分和机械部分——是有可能的，装配所需要的几乎所有的部件都可以在网上订购，然后可以找一家低成本生产商生产出设备原型。组织这样一个研发设备的在线开源项目并非遥不可及，而且它也不需要数百万美元的启动资金。

然而这样设计出来的设备能否跟那些财力雄厚的大制造商所研发的设备成功竞争呢？Gartner 专门研究开源软件的分析师 Mark Driver 认为，开源器具的最好机会就在那些技术已经相当成熟的市场中。

“开源器具主要面向商品化的市场，” Driver 说。“这些产品所处的市场实际上没有太多实际上的差异，彼此间可以相互替代，这应该就是开源器具的用武之地。Linux 之所以获得广泛的欢迎，并不是因为它有什么新功能，而是因为它能做 Unix 能做的一切，但是采用了更开放的方式。如此而已。”

开源硬件的定义

虽说如今市面上已经有了数量庞大的开源电脑和电子器件，但是却只有为数不多的科技器具是完全按照开源哲学进行研发的。要准确地将某个硬件项目定义为开源项目，最重要的定义项还是开放。

一般而言，“开源的”硬件至少是指其研发计划的某些部分是完全公开的，可允许其他人对此研发项目做出贡献，或者经其创建者的允许可自行生产设备，甚至修改研发计划以开发新的设备。例如，Always Innovating 公司就鼓励外部人士对其基于 ARM 处理器的平板电脑/上网本混搭产品——Touch Book(触摸本)的设计做出贡献。该触摸本仅重 1.8 磅，配有一块触摸屏、可拆卸键盘和用户定制的 Linux 操作系统。单块电池的续航能力长达 10 小时。

该触摸本的原理图可从 Always Innovating 的网站上自由下载。“我们还会为那些以我们的设计为起点，开发他们自己设备的企业提供更大的支持和咨询服务，” Always Innovating 的 COO Alexandre Tisserant 说。

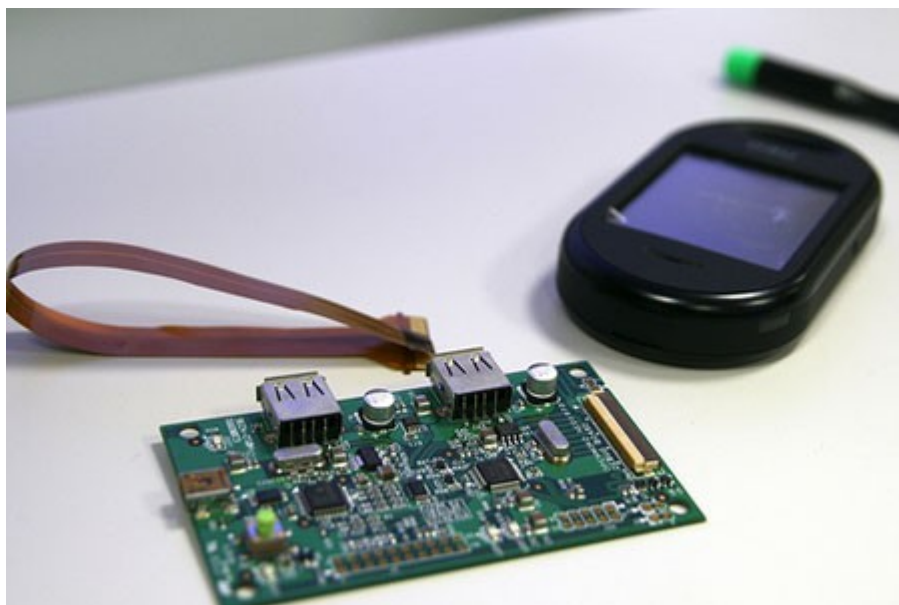
“我们所遵循的原则是：开发可靠的、创新的产品，并通过开放这些产品，获得所需的回馈和建议，进一步改进产品，设计出更快更简便的新产品，” Tisserant 说。这正是开源的理念。从不利的一面看，“最糟糕的情况可能是，一个新的项目挂着开源的名头，但最终证明不过是一个营销噱

头而已，” Gartner 的 Driver 说。“如果开源设计项目只出自一家厂商，或只有一个支持源，这种情况就是最不充分的开源形式。”

谁是开源硬件市场的顾客？

毫不奇怪，这类器具主要面向(或者主要诉诸于)科技爱好者。Touch Book 就主要卖给这群人，Tisserant 说，到目前为止已经卖出了“数千部”。不过该公司也在寻找一些垂直市场。因为 Touch Book 是高度可定制的，很容易将其集成到出租车或者警车上，或者连接到医院的专用网络上作为医护人员使用的一种“永远在线”的便携设备。

另外一个例子就是开源相机 Frankencamera，这款基于 Linux 操作系统的数码相机可以通过编程来控制曝光、闪光、聚焦设置和其他功能。该相机由斯坦福大学的一群学生开发，主要用于学术研究。Frankencamera 的一位带头开发者 Andrew Adams 说，“具体来说，我们就是希望可能会用到编程相机的研究生或者上编程课的本科生用起来更方便。我们自己就是研究生，这整个项目就是我们千方百计想让原来的编程相机完成我们想要完成的作业失败之后开始的。”



迄今为止，在面向消费者的开源项目上还没有哪个项目可以赶上 Neo FreeRunner 智能手机项目，该手机支持 Linux 平台，研发代号叫做 Openmoko，是由 Openmoko 公司发布的。它的操作系统和内部电子线路设计方案任何人都可以使用，并可对其继续改进。该公司在 2009 年 4 月正式停止了对这个项目的支持，该公司的产品经理 William Lai 说，“随着时间的演进和技术的进步，要想和苹果、RIM、Android 等进行竞争所需要的投入已经超出了我们的设想，我们很快意识到，要想按时完成这项技术已经超出了我们的能力。”

不过，Openmoko 平台和 FreeRunner 手机仍然在由一个志愿者社区继续进行开发。

硬件的分发与测试均很困难

对于软件而言，任何人都可以下载一份开源软件的拷贝，并立刻投入实际使用。向软件的开发者提供反馈，贡献漏洞修补代码或增加功能等，也同样的容易。

软件开发的卡开源模式就是在这样一种连续不断的贡献与测试的进程中繁荣起来的：能够分发给其他人的拷贝越多，项目就会进展得越快，开发人员也就能更快地修正错误，改进软件，从而进

展到下一个版本。

然而要将开源模式应用于硬件却并非如此直接。要生产和分发设备原型的“拷贝”，提供给其他开发人员用于评估和测试，这其中的成本是相当昂贵的，所以整个开发进程不可能很快。

Tisserant 称此为“测试成本”。“当你在做第一件自制硬件时，你可以在上面做某些修改。但是有了新的设计，就需要制作新的硬件。这可是需要时间(数周)和钞票的。”

Tisserant 认为，要想真正地向硬件研发的传统专利模式发起挑战，原型的制作周期就不能长于一周。“研制和测试越便捷，你就会学得越快。”

使用可负担得起的快速原型或成型机械，可以减少设计-原型-测试的周转时间。例如 Frankencamera 的机身就是可激光切割的丙烯酸塑料。任何人都可以利用一台激光切割机来修改 Frankencamera 机身的设计，制作他们自己的相机。像 RepRap 这样的 3D 打印机就可用于快速原型设备制作，在开源硬件的研发中发挥重要作用。RepRap 本身也是开源的。低端的商用 3D 打印机的价格一般都在 2 万美元左右，而 RepRap 的设计却是任何想要自己造一台的人都能免费得到的。这种 3D 打印机的制作成本约为 480 美元。更有趣的是，RepRap 可以复制装配自己所需的很多部件，其他部件可以很便宜地买到，所以你可以用第一台打印机去制作另外一台。

“用 RepRap 或激光切割机和烙铁，有的人就可以制作硬件了。开源硬件设计加上快速制造，就能精确复制开源软件的成功——只要设计是开放的，你就可以修改它以满足你的要求，并可与他人自由地分享这些修改，” Adams 说。虽然 RepRap 和其他快速原型机可加快原型设备的制作过程，但这些东西并非圆满的解决方案，因为它们的能力也仅限于制作器具的外壳。此类机械可帮助快速制作比如说上网本的外壳，但上网本的大部分内部电子器件仍需要外包给专业的制造商。

不过，把一台设备的研制过程开放给公众(尤其在其早期设计阶段)，有助于促成一个社区的形成，该社区会提出修改建议，或者直接贡献修改设计。这将会有助于减少制作原型设备的数量，节省金钱和时间。

零件厂商缺乏开源文化精神

一台开源的设备，并不意味着其内部每个零件的设计都是开源的——事实上，其内部零件可能使用的是一些专利的零件。任何消费者科技设备都是由很多更小的零件组装成的。而这些零件的生产商们通常都不愿意公开其生产流程，除非你花钱委托他们生产。这一点对于任何想要研制开源硬件的人来说都是一个棘手的问题，假如你想要公开设备的设计方案的话。

“在软件世界，存在着提供基本开源开发模块的丰富文化，比如提供编译器、编辑器、支持库和操作系统，” Frankencamera 项目的 Adams 说。“但是很遗憾，芯片制造是一种本身就投入很大的业务，所以很少会有像开源贡献者那样无私共享的想法。你得签署保密协议才能看到某个零件，比如传感器是怎么制作的。”

尽管 Adams 及其 Frankencamera 项目的开发者们经常会遇到帮助他们研制数码相机零件的一些厂商不情愿或者拒绝透露零件的生产流程和相关信息，但是他们还是得到了一些厂商的奉献——这主要是因为这一项目的开源特色。Frankencamera 数码相机所使用的大部分电子元件都是商品化的，什么人都可以买得到。还有一些部件，比如相机的电源部分就是由该项目设计团队设计的。

“要想让企业公开零部件的生产制造细节很困难，他们并不在乎你是要制作开源硬件还是商用硬件，同样都会保持沉默。而愿意提供帮助的人和企业一般更多的是冲着开源项目才提供帮助的，因为他们知道，他们也能从最终的结果中获得好处，” Adams 说。

知识产权的问题

困扰着开源硬件项目的一个大问题便是知识产权的法律问题——谁拥有开源设备，尤其是当有好几个人都对该设备的设计做出过贡献的情形。Brenda Scott，一位专门从事 IT 业法律案件的律师，并运行着一家叫做开源法律(Open Source Law)的网站，他强烈建议开源硬件项目的创建者和主要开发者必须在任何人同意加入该项目之前便解决好知识产权的问题。

至于如何解决这一问题，他则认为并不存在千篇一律的答案。“在某些场合，个人最好保留知识产权；而在另一些场合，则最好将知识产权转移给某个持有实体。关于开源项目的知识产权，最重要的是要在项目开始之前——或者项目刚开始之时——便考虑这一问题，而不要等到项目完成之时再去考虑。如果不解决这个问题，你可能会发现，这个问题就会决定你的未来，而且很可能使用一种让你很不愉快的方式。”

TechCrunch 博客的创办者兼共同编辑 Michael Arrington 也会赞成这一点。2008 年 7 月，他宣布了创建一个低成本 Web 平板的计划，也就是后来叫做 CrunchPad 的项目。尽管这个硬件研发项目并未完全开放，但是 Arrington 的想法就是“设计它，生产几件，然后将研制方案开源，以便任何人都可以修改它们，”他在宣布计划时就是这么说的。该项目在 TechCrunch 与新加坡的 Fusion Garage 开始合作研发并制造 CrunchPad 之后有了一个很不错的开端。然而到了 2009 年末，当 Fusion Garage 宣称它不必经过 TechCrunch 同意可销售 CrunchPad 时，双方的合作破裂了。Fusion Garage 的 CEO Chandra Rathakrishnan 声称他的公司唯一拥有该设备的知识产权，而 Arrington 则坚称应是双方共同拥有该设备的知识产权。

Fusion Garage 计划将此设备以 JooJoo 平板电脑的名称予以销售，而 TechCrunch 则将 Fusion Garage 告上了法庭。在本文截稿之时，Fusion Garage 已开始办理 JooJoo 的预定，该公司的网站称，现在预定“将会在 8 到 10 周内供货”。

在问到 Frankencamera 研发项目所采取的法律步骤时，Adams 说，“在这方面，由于没有牵涉到钱的问题，所以生活还是很轻松的。因为我们所做的每件事都是在上学的时候做的，如果有人想做邪恶之事的话，我们也有很好的司法途径可走。所以到目前为止，我们从一般公众那里所听到的都是关心、鼓励和帮助。”

现有开源许可证可适用于硬件

另一个法律问题是，现有的开源许可证可否适用于开源硬件，或者至少是足够适用的。这些许可证的大部分条款都是涉及软件的，这一点在其措辞中很明显：普遍使用的 GNU 通用公共许可证就主要涉及“程序”。

开源法律网站的 Scott 曾写过一篇内容丰富文章，他就认为，现有的开源许可证不必做任何修改便可适用于开源硬件项目。“把硬件设计和零件清单视为某个‘对象’(从字面上讲)的源代码，可‘编译’或者‘组装’该对象，这并不是很牵强，”他说。

Scott 预计，随着时间的推移，将会对开源许可证进行定制或修改，以便覆盖开源硬件所遇到的问题。例如，TAPR 开源硬件许可证就是专门为开源硬件项目而设计的。而 Arduino 项目，一个开源的电子平台，它既包含硬件也包含软件，它的硬件设计所用的许可证和固件(操作系统在其上运行)的许可证是不同的。

开源硬件如何赚钱

Always Innovating 的 Tisserant 承认，硬件企业如果走开源路径，可能会降低利润率，但是他说，企业可以从较低的 R&A 成本和更短的开发周期中获益。“开源的目标不是要保住你的秘密，然

后靠着无穷尽的版税收入生活，而是要分享知识，从而更更快速地促进创新，”他说。

虽然 Openmoko 公司不再支持 FreeRunner 手机和 Openmoko 智能手机平台，但是 Lai 称，公司并没有放弃开源模式：“去年，Openmoko 一直在把各种开源要素介绍给公众。我们会继续利用开源要素去设计产品，”例如 WikiReader(售价 99 美元)，就是一款预装了 Wikipedia 内容的开源口袋阅读器。

至于对 Frankencamera 的研发团队来说，他们并没有什么商业计划，因为他们的项目并不打算销售最终产品。其目标只是要把 Frankencamera 的电路原理图开放给其他学术机构的学生，以便学生们能够用最少的成本制作自己的数码相机，用于课程学习和学术研究。

其次，他们希望自己的项目能够“让相机制造商们看到，允许最终用户对其数码相机进行编程，是一件可以实际增加产品价值，让更多人使用其产品的事情，因为一个爱好者社区会不断地会产品添加新的功能，”Adams 说。

“如果没有应用商店，iPhone 会如此成功吗？那么你为什么不能编写和下载应用到你的相机上呢？我们的目标就是要做这样有意义的研究，为其他人提供做这种研究的工具，而不是为了赚钱，”Adams 说。

向极客圈外销售开源硬件

Jeff Orr 是 ABI 研究公司的技术分析师，他认为开源硬件要想在市场上获得成功，与专利的商用产品开展竞争，就仍然需要有“某个所有者——无论是个人还是实体——来负责组装和分发这些产品。否则得话，消费者购买了开源硬件产品后，由谁来提供支持呢？一旦出了问题，由谁来保修呢？”他对尚处于研发阶段的开源硬件的前景表示出谨慎乐观的态度：“开源模式能否挑战商业研发模式呢？我认为可以，因为开源模式创造了一个范围更广的知识库，任何个人或组织都能够从中学到东西。”

开源硬件能否会像 Firefox 和 Ubuntu 那样，以同样的方式起步，在主流的消费者器具用户群中成为一个家喻户晓的名字呢？Gartner 的 Driver 预测，开源硬件器具将会越来越普及，但是他不能肯定，业界是否会看到一种有着广大用户基础的开源硬件需求，能够向专利硬件产品发起挑战。“我们是否能看到像在软件领域一样的一场开源硬件的革命呢？我认为可能性不大，至少在可预见的未来是如此，”Driver 说。

社区扫描

Google 赠送手机给 Android 开发者

Google 今天通过电子邮件的方式宣布了一个新的开发 Android 应用程序送手机的活动。作为开发者，只要你的应用程序在 Android Market 中的评分超过 3.5 颗星，以及达到 5000 次的单独下载量，你就可以获得一部免费的摩托罗拉 Droid 或者 Neuxs One。但开发者不能自己选择你要哪一款手机，而会由你所在的国家和其他因素来决定。

不知道到底多少开发者符合条件，但估计 Google 应该需要至少准备几千部手机。

很多开发者在收到邮件时都持有怀疑的态度，毕竟天上掉馅饼的事比较罕见。Google 的 Android 开发者关系代表 Roman Nurik 在 [Android Developers Google Group](#) 这样回应大家的疑问：

“一个欣欣向荣的开发者社区是为来自全球的用户创造出更好的移动体验的一个重要的组成部分。我们希望通过为开发者提供手机，能让他们开发和测试应用程序更方便一点。我们还准备了与此关联的其他活动来为开发者提供支持，包括我们的 Android Developer Labs World Tour 和参加即将举行的 Game Developers Conference。”

GNOME 3：面向“Task”的桌面

由 Ubuntu 背后的公司 Canonical 和 Google 赞助进行的《[GNOME Usability Hackfest 2010](#)》已经在圆满结束，关于 GNOME 3 的更详细特性已经显露。

开发者已经放出 GNOME 3 UI 交互设计的草稿，顿时掀起一阵热烈讨论。

著名的科技网站 Ars Technica 写了篇关于 GNOME 3 的专题报道：[“Task Pooper” could revolutionize GNOME desktop](#)，这篇文章将开发者放出的草稿做了自己的总结和理解。后来还被转载至了 LWN。

看来 GNOME 3 的一些概念还是值得分享的，那么到底是怎么样的概念呢？下面是我的理解。

GNOME 3 将引入“Task Pooper”概念，它将用户在桌面操作的对象，从窗口或文档，转为“Task”。一般情况下，Task 指的也是应用程序的窗口，但它也可以是整个工作区。

想像一下，你正在“工作区 A”编辑文档，可能打开了两三个相互辅助的应用程序。这整个工作区及里面的应用程序的状态，构成了一个 Task。

你可能要花一周的时间（每天几小时）来编辑这份文档。这期间，电脑总不会一直开着吧？要么休眠，供下次快速调用，要么就关机，下次再一个一个调用起来。

在 GNOME 3 当中，不需要这么麻烦。假设今天的编辑任务已经完成，你不需要保存文档，然后关闭应用程序。只需要将整个“工作区 A”，拖至标签为“Tomorrow”的 Task Pooper 中。然后，你可以不管之前的任务，尽管去做其他事情吧。其他的 Task，也可以用类似的形式归档。供后期调出。

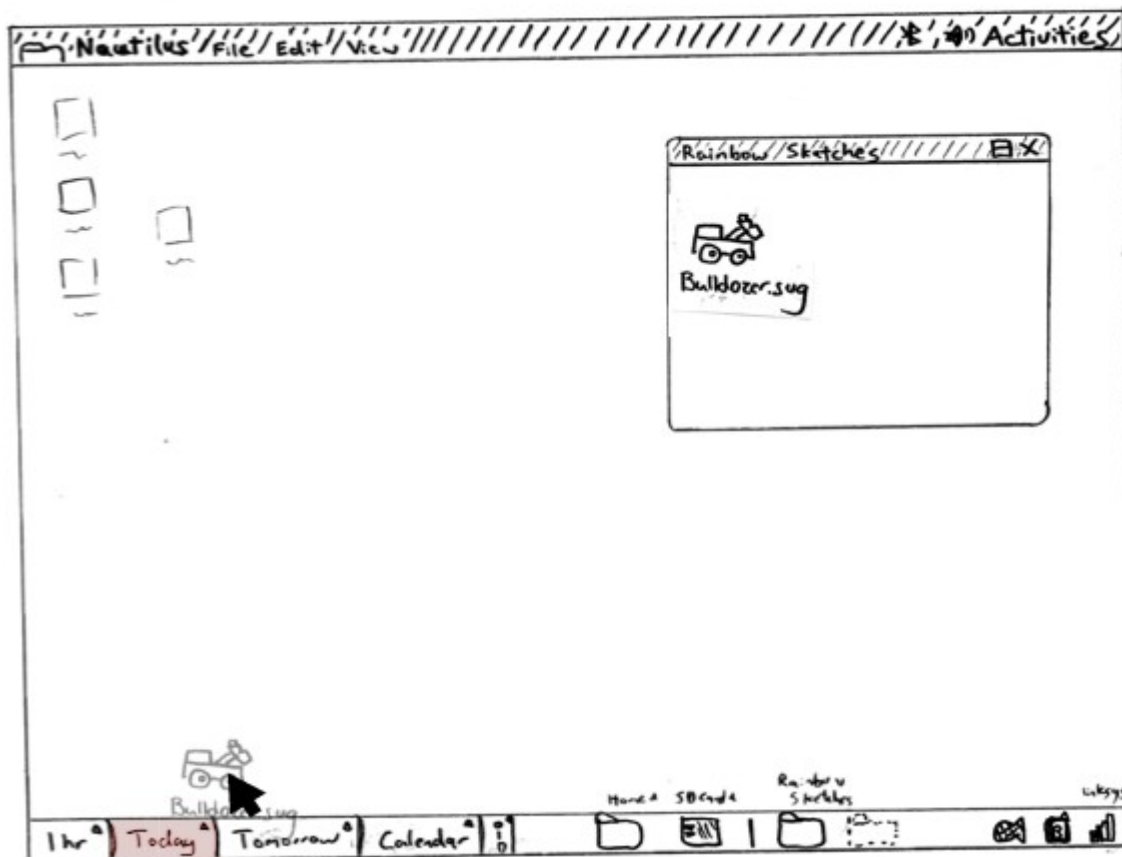
这解决了用电脑时，每天重复地打开、关闭一些常用应用程序的动作。对于经常有多个不同的

Task 需要处理的人来说，这个机制能帮助他很好的归档 Task。

这个概念跟当前 GNOME 桌面已有的“保存当前会话类似”，但是当前的是针对整个会话来说的，而不能分门别类和灵活的调用。

在文章中，并没有解释处理保存应用程序当前的状态的机制。应该是会在桌面底层来增加支持这个特性的机制吧。

最近附上一张概念图，图中的 Nautilus 居然是 Global Menu 的，难道 GNOME 3 真的要采用这个设计？



关于新概念的解读和理解可能有不少个人因素或偏差，欢迎大家针对未来的 GNOME 3 进行讨论！

Digg 用 NoSQL 替代 MySQL

继 Twitter 之后，社交新闻网站 Digg 决定跟 MySQL 说再见，并替换掉它的大部分基础设施组成，Digg 将从 LAMP（Linux、Apache、MySQL 和 Perl/PHP/Python）架构迁移到基于 Cassandra 的 NoSQL 架构。

Apache Cassandra 是一套开源分布式数据库管理系统，最初由 Facebook 开发，用于储存特别大的数据。Cassandra 是一个混合型的非关系的数据库，主要特点是它不是一个数据库，而是由一堆数据库节点共同构成的一个分布式网络服务，对 Cassandra 的一个写操作，会被复制到其它节点上，对 Cassandra 的读操作，也会被路由到某个节点上面去读取。

Cassandra 的使用者现在包括了 Rackspace、Digg、Facebook、Twitter、Cisco 等等。

Mozilla 复查其开源协议 或与 Apache 协议兼容

Mozilla Public License (MPL)，也就是 Mozilla 管理旗下软件开源的许可协议，上一次修订还是十多年前的事情。周三 Mozilla 发布公告称，将对这一长寿的许可协议进行复查和部分修正。这一协议最大的问题在于，在允许人们修改源代码的同时，允许企业和其他人分发软件的二进制成品——这意味着将 MPL 代码与自家的闭源程序混合，违背了开源精神。Mozilla 目前已经放出公开讨论版提供意见平台，并计划在今年 10-11 月完成修改计划。新的协议依然将遵循自由软件基金会对于“自由”的定义，并符合开源倡导者们对“开源”的定义，同时维护协议本身的简洁。Mozilla 表示更多可能被纳入修改计划，但不保证的新变化有：

兼容 Apache（及其协议），在 Mozilla 的专利和其他项目中更平滑的使用基于 apache 协议的代码。

Mozilla 已有的专利将继续向上兼容其可用性。

将协议全球化。

将协议变成类似模板的样式，防止协议在闭源代码中蔓延滥用。

在新开发的项目中区分源代码和可执行程序的适用范围，包括编译语言，二进制代码修正，针对非代码用户等等。

这不是第一次有人希望对 MPL 进行修改。Sun 曾经发出过“通用开发和分发协议”（CDDL）的倡议，但是 Mozilla 认为改动太大，没有接受。本次修改中值得注意的是和 Apache 这一应用广泛的开源协议的兼容，这将提高 MPL 协议下代码的可用性。本次修改由 Mozilla 主席 Mitchell Baker 领衔。

微软调整业务战略 不再敌视开源社区

微软美国市场技术高管斯图亚特·麦基(Stuart McKee)周四表示，微软已经调整了此前对开源社区的敌视态度。

麦基周四承认，微软此前与开源界存在对立关系，但如今已经调整了这种关系。最近几年，微软曾公开拥护开源应用，甚至还资助过一些开源运动。麦基说：“我们已经意识到，微软的成功必须基于这样一个事实：在微软的平台上可以运行不同的技术，其中包括开源。”麦基接着说，微软曾经与开源支持者存在一定争议关系，但这种关系真正在改变。

微软早在 2007 年曾表示，开源软件侵犯其 230 多项专利。其中，Linux 内核侵犯 42 项专利，Linux 用户界面和其它设计元素侵犯 65 项，OpenOffice.org 侵犯 45 项，其它开源软件侵犯 83 项。微软此举在当时引发业内恐慌，同时也遭到了开源社区的一致谴责。

Eric Raymond：几种计算机语言的评价

编程新手都有一个同样的问题：我应该学习哪一种语言？

《Unix 编程艺术》（Eric Raymond 著）第十四章，对各种语言进行了评价，正好可以用来回答这个问题。下面是知名 IT 博客阮一峰的网络日志，整理了六种主要计算机语言的优缺点。读完就会知道，对于不同的任务，应该选择哪一种语言了。

=====

几种计算机语言的评价

作者：Eric Raymond

摘自《Unix 编程艺术（第十四章）》



一、C

C 语言的优点是，运行效率高和接近机器语言。它特别适用于以下几种程序：

1. 对运行速度要求很高的程序。
2. 与操作系统内核关系密切的程序。
3. 必须在多个操作系统上移植的程序。

除此之外，学习 C 语言有一个最大的理由，那就是它能帮助我们学会，在硬件层次上考虑问题。如果你想以编程作为自己的人生事业，C 是一定要学习的。

C 的最大缺点是，在编程过程中，你不得不花很多时间，考虑与你要解决的问题完全无关、且非常复杂的硬件资源管理问题。

二、C++

C++ 在 80 年代中期推出，支持 OO（面向对象编程），原意是作为 C 语言的取代者。

但是它没能做到做一点，原因有以下几个：

1. 它也没有解决内存管理问题，虽然比 C 有所改善。
2. 标准化不成功。各个编译器都只支持自己选择的一个子集，导致跨平台性不如 C。
3. 过分的精细和复杂了。C++ 的复杂度，超过了 C 和 OO 的复杂度之和。
4. OO 并没有带来多少优势，反而带来了一些副作用，比如厚重的胶合层和庞大的代码体积。

总的来说，C++ 的优势还是程序效率，以及面向对象编程能力，糟糕之处是它鼓励复杂的设计。

三、Shell

Shell 程序写起来很容易，对于简单的任务，开发速度很快。

当程序变大时，它就不太适合了，很难维护，而且将变得非常专用（只能在你自己的机器上使用），因为 Shell 必须调用各种外部程序，无法保证每一台机器都同样安装了这些程序。

最常见的 Shell 是 bash，它的一些语法规则很混乱，会产生很难阅读的代码。另外，shell 只能在 Unix 上使用，无法跨平台。

Shell 的最佳用途如下：

1. 执行简单的系统管理任务。
2. 编写系统启动时的初始化脚本。
3. 作为其他语言开发的程序的包装器。

四、Perl

Perl 发布于 1987 年，基本上就是一个增强的 Shell。它的最大长处是强劲的文本处理能力，无以伦比的正则表达式支持，而且有全套 Unix API 的内部支持，显著减少了对 C 的需求。

Perl 的主要缺点是某些部分设计得非常丑陋，且无法补救，某些部分也过于复杂。当程序规模增大时，必须严格遵守约定，才能保证模块化和设计的可控性，难于维护。

五、Python

Python 发布于 1991 年，是一种与 C 语言紧密集成的脚本语言。Python 的优点是非常干净优雅，鼓励清晰易读的代码，易学易用；提供了面向对象编程的可能，但并不把这个选择强加于设计者；具有出色的模块化特性，同 Java 一样，适合用来做需要协同开发的大型复杂项目。在很多方面，它都比 Java 简单。此外，Python 标准中包括了对许多网络协议的支持，因此也很适合网络管理任务。

Python 的缺点主要是效率低下，速度缓慢。在执行速度上，它不仅无法与 C/C++ 竞争，而且也不如其他主要的脚本语言。但是，这其实并不是一个严重的问题，有时网络或磁盘的延迟，会完全抵消 Python 本身消耗的时间。而且因为 Python 特别容易和 C 结合起来，因此性能关键的 Python 模块，可以很方便地转化成 C 语言来提高速度。

总的来说，对于小型项目和大量依靠正则表达式的项目，Python 不如 Perl 的表达能力强。至于更小的项目，Python 则是大材小用，shell 也许更适合。

六、Java

Java 发布于 1995 年，设计目标有两个。一个是 write once, run anywhere（一次编写，到处运行），即不依赖于特定的平台；另一个是在网页中嵌入交互程序（applet），可以在任何一个浏览器中运行。由于它的所有者 Sun 公司的一系列失误，第一个目标并没有完全实现，第二个目标则是彻底失败。但是 Java 依然在系统编程和应用编程方面非常强大，足以挑战 C 和 C++。

Java 的优点是比 C++ 小巧简单，可以自动管理内存，支持类似 C 的语法和 OO 编程，与 C 程序的结合也很好。Java 的缺点是某些部分过于复杂，比如内部类和匿名类的运用会产生很混乱费解的代码；某些部分功能不完善，也无法利用操作系统提供的功能接口，比如在 Java 中读取和处理文本文件，并不像其他语言那样容易。此外，Java 配置环境和版本的混乱，也让人很头疼。

总的来说，除了系统编程和某些对运行速度要求很高的编程之外，Java 都是比 C++ 更好的选择。如果和 Python 相比，Java 可能在大型项目上有优势，但是也不是绝对的。

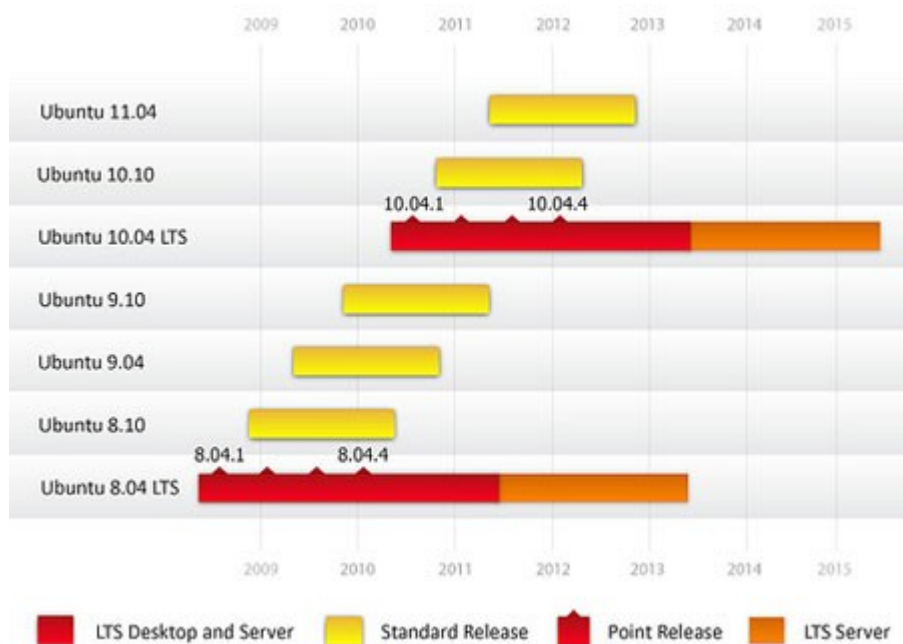
Nvidia 放弃对开源驱动的支持

虽然 Nvidia 一直对自由开源世界不太友好(尽管这里已经口水战很久了),但他们还是支持着可以提供基本的模式设置和一些基本功能的 xf86-video-nv 显卡驱动。

但"Fermi"和未来的产品都将停止对现有开源驱动的支持.Nvidia 宣布他们放弃所有的未来 GPU 产品的开源驱动的支持.Nvidia 建议去官方网站下载通用的 X.Org Versa 驱动,这是一款私有软件的驱动。今天的 GNU/Linux 用户大部分还是使用的这一驱动.幸运的是仍然有 Nouveau 项目作为支持 2D 和 3D 的针对 nvidia 硬件的驱动,但 Nvidia 对此不感冒也不会对此有任何官方的支持。"

Ubuntu 10.04 测试版 加入社交功能

代号 Lucid Lynx 的 Ubuntu 10.04 LTS 推出测试版,系统直接连结了 Facebook、Twitter 等 9 种社交服务。如其他的 Linux 版一样,新版本 Ubuntu 主要是升级 Linux 核心、GNOME 桌面、Firefox 浏览器...等套件为基础。此外,Canonical 公司的 Ubuntu 针对企业对云计算的需求,改进企业私有云软件 Ubuntu Enterprise Cloud (简称 UEC) 的功能,并增加 Ubuntu One File Syncing 这项档案同步功能。



在桌面方面,主要通过 GNOME 的更新,提供一个更加抢眼、便利的操作环境。新版本也整合 Gwibber 社交网站整合专案,以 MeMenu 面板连结呈现 Facebook、Twitter 等 9 个社交网站的讯息。特别值得注意的是,Canonical 公司这次搭配 Rhythmbox 音乐播放程式,跨入线上音乐商店的市场。

LTS 是长期支援的意思,一般的 Linux 通常仅提供半年的支持,而对于 LTS 版本 Canonical 公司至少会提供桌机版三年、服务器版五年的支援,适合企业搭配服务器、企业日常作业等需要长期技术支持的电脑使用。Ubuntu Linux 自 2006 年开始针对企业推出 6.06 LTS,2008 年推出 8.04 LTS,最新的 10.04 LTS 测试版已经可以下载,正式版预计在今年四月 29 日推出。

行业观察

Linux 是否已经迎来了自己盈利的春天

关于 Linux 的盈利问题，一直是业界人士关注的焦点。一开始，业界人士对涉及 Linux 技术的公司有一种疑问：免费的东西如何实现盈利？好在这个问题早在 Linux 发展最困难的 2010 年，被 Red Hat 公司的盈利结果击得粉碎。Red Hat 用自己的财报在当时向世人宣告了：免费的东西，通过提供不免费的服务，一样可以实现盈利。这个观点到了今天，已不再新鲜了。因为软件已在向服务转化，当我们把目光转向未来时，我们甚至已经发现：服务已在威胁到软件的生存了。

近日，Linux 业界的另一个巨头 Novell 又给了我们一个惊喜。在上周 Novell 发布的截至 1 月 31 日的 2010 财年首份季度收入报告中，公司的营业额为 2 亿 200 万美元，比起 2009 年同期的 2 亿 1500 万略有下滑。而在净收入方面的情况好了许多，Novell 报告说公司的净收入为 2000 万美元或每股 0.06 美元，比起去年同期的 1100 万美元或每股 0.03 美元有了很大幅度的增加。Novell 首席财务官 Dana Russell 介绍说：“Linux 平台的产品收入增长了 6%，达到 3700 万美元，并且取得了收支平衡。”

2003 年，Novell 以 2100 万美元收购了 SUSE Linux 开始进入 Linux 业务。一直到 2009 年 5 月，Novell 的 Linux 的业务依旧无法赚钱，但 Dana Russell 当时已经敢于宣称将在未来 12 至 18 个月的时间取得收支平衡。而上周 Novell 发布的财报印证了 Dana Russell 当年的誓言。这样，从 2003 年到现在，七年的时间过去了，Novell 终于第一次实现了收支平衡。

这些业绩确实可以说明 Linux 在需要服务的企业级，确实已经迎来了自己盈利的春天。但另一方面，我们必然看到，这种依告服务的模式并不适用于桌面端 Linux。

原因很简单，因为在桌面端，个人可能因为觉得太麻烦而不愿付费，个人使用的软件也没必要复杂到请人来做服务的地步。我们可以看到，不少世界级的桌面端软件并没有实现盈利。

但如果因此就放言桌面端 Linux 不能盈利，未免目光过于短视了。

一方面，Linux 的竞争对手正在收缩战线，这极可能给桌面端 Linux 以机会。例如微软已经宣布：微软将于 2010 年 4 月 13 日停止对未安装任何 SP 包的 Windows Vista 的支持。这意味着微软将不会为该版本 Vista 提供任何安全更新，不过安装了 SP1 和 SP2 的用户仍将继续获得微软的支持。此外，微软将于 2010 年 7 月 13 日停止对 Windows XP SP2 和全部版本 Windows 2000 的支持。

另一方面，不断变化的硬件产品也在为 Linux 提供机会。上海中标软件有限公司总裁韩乃平就认为：“在传统 PC 生态圈，Windows 应用充斥市场，数码相机等各种产品的驱动程序都是为 Windows 而设，但在上网本这个新市场就有所不同，在这个市场里，没有传统应用的包袱。”这实际上给桌面端 Linux 提供了机会，具体到盈利层面，这就意味着：除了传统的 Linux 盈利模式，在上网本市场以及手机市场，出现了第三方付费方式，运营商卖上网本也好，或者是 PC 厂商直接卖硬件也好，都需要包含应用在内，而且这种应用不仅仅只有操作系统，还需要包括配套的端到端的解决方案。

而说到桌面端 Linux，最有发言权的莫过于 Ubuntu 的创始人 Mark Shuttleworth 了。由于 Linux 服务器市场现在是红帽子和 Novell 公司的地盘，所以 Ubuntu 只好把自己的着眼点放在桌面端。

谈到盈利问题时，Mark Shuttleworth 认为：“我不认为通过销售桌面版 Linux 能挣多少钱。但是通过它有机会创建一些伟大的事情，而且我们也做到了。”事实上就是，当 Novell 和 Red Hat 最终放弃了桌面版 Linux 的努力时，Ubuntu 却拿到了市场主动权。这为 Mark Shuttleworth 争取到了一个极为庞大的用户群，这个用户可以使 Mark Shuttleworth 干很多事情。

这并不是空谈，我们可以看一下 FireFox 的盈利模式，说白了，FireFox 的盈利模式就是和 google 搜索整合到一起，从 google 分到利润。同样的事情，基于 Ubuntu 的用户群，Mark Shuttleworth 肯定也能办到。

由此不难看出，在服务器和桌面端的 Linux 挣到钱以后，接下来，桌面端 Linux 的盈利期实际上已经来到了。但是否能真正化机会为实际的财富，还要看桌面端 Linux 把握机会的能力。

发展 Linux：中日两国之比较

大约是在 1997 年的夏天，国内召开第一次 Linux 高层邀请研讨会，我想，现今那次邀请会的全体与会者都还健在，姓名我就不说了。记得，我去机场迎接 Cliff Miller 夫妇，那时，Cliff 是日本 TurboLinux 的 CEO，专程来京参加这次 Linux 邀请讨论会。会后，我在某专业 IT 报纸上发表文章，题目是“向右，还是向左？”（Copyright, or Copyleft?），阐明了 GNU/Linux 的发展路线。

1998 年，日本启动一项“Project Japanese Extension”计划（又称 JPE 计划），建立自由软件的“Vine Seed”（“葡萄种子库”），类似现在的“自由软件源”（Repository），计划发展一种称为“Vine”的 GNU/Linux 日文发行版，其中不含任何私有软件。该项目坚持至今，并于去年 8 月 25 日发布了 Vine 5.0 版本，进入了全球先进行列，日本走上了发展 Linux 的 Copyleft 道路。

反观我们国内，发展 Linux 走的却是 Copyright 道路，比如，蓝点、冲浪，.....其主要特征就是主张发展“国产 Linux”，全面对抗微软。至今，无人能够站出来否认“国产”Linux 缺乏 GNU/Linux 的本质特征（即 Copyleft）。实际上，“国产”Linux 的代表作就是红旗 Linux 中文发行版。



在全球唯一的 Linux 发行版排行榜（www.distroWatch.com）上，如果统计间隔为一个月的话，那么，我们可以看出，Red Hat 排名 25，Vine 排名 44，Moblin 排名 61，红旗 Linux 排名

213, 以及 Asianux 排名 231, (红旗 Linux) 落伍何止千米。很明显的事实是, 日本 Vine(即 GNU/Linux) 日文发行版的 Copyleft 路线胜出。

现今, 中、日两国都是全球经济发展中的经济大国, 我们还是发展中的国家。实际上, 中国人口要比日本多, 理应要有一个像样的 GNU/Linux 发行版。有人说我左、右摇晃(一会儿 Copyleft, 一会儿 Copyright), 但是, 认识我的人说我还是老样子。昨天, 我在 csdn 网站上正好发表了 1,234 篇短文, 前后观之, 虽然似乎存在矛盾。不过, 仔细观看每篇文章之要点, 不外 Linux pk Windows。近期, 我主张走 Live 中间路线, 不左不右。

在 www.osdisc.com 网站上, 有 Vine 5.1(体积 1MB 多一点儿)的 LiveDVD 光盘出售, 5.95 美元一张, 销售排名第 39。在 Vine 5.1 发行版的主页面右下方有一束含苞欲放的樱花伸出, 在页面左上方, 下拉菜单的条目里面全是日文, 怪模怪样, 也很可爱。GNU/Linux 中文发行版, 你在哪里呢?

打折出售: Linux 厂商 Novell 将花落谁家?

纽约机构出价收购 SUSE Linux 的母公司 NOVELL 的消息在企业级 Linux 市场上引发广泛关注。无论本次交易成功与否, NOVELL 的命运都让人有些怀疑, 根据过往的经验, 我们可以诚实的说, 没有人还会去购买 NetWare 系统。

美国私募基金 Elliott Associates 本次收购要约报价为每股 5.75 美元, 对 NOVELL 市场估值不到 20 亿美元。但需要谨记的是, 公司拥有各种币种的充足货币储备, 价值约为 9.5 亿美元, 收购要约对 NOVELL 的账面价值约为 10 亿美元。

Elliott 能从这 10 亿美元的业务估值中得到什么呢? 当然是 SUSE Linux, 还有 NOVELL 旗下的各种管理, 认证和安全产品。因此得到的决不仅是只一家 Linux 公司。NOVELL 在这些领域确实占有一席之地, 但是单就 Linux 市场来说就已经是大家争夺的热点。

首先来对比一下 NOVELL 和其他的大型企业级 Linux 厂商: 红帽和他们旗下的红帽企业版 Linux 操作系统。红帽不仅是一家 Linux 的制造商, 还涉猎以 KVM 为基础的虚拟化领域, 更不必说 Jboss 中间件产品。红帽目前的市场股价约为 30 美元, 公司市场估值超过 56 亿美元。现金流和等价物价值约为 9.5 亿美元。使得红帽的账面价值超过 45 亿美元。

因此对于每个有意收购企业级 Linux 制造商的买家来说, 对 SUSE 的投标起价约为 10 亿美元, 对红帽的出价会超过 45 亿美元, 可能会接近 60 亿美元。对比之下, SUSE 看起来还是很便宜的。在对 Linux 的争夺中, 多数人认为红帽要更胜一筹, 但是正如价值差异所显示的, 再好能好出六倍以上吗?

这场 Linux 领域的座次之争是否会让 Elliott 的出价收购激发更多的厂商加入这场争夺。有行业观察家提出了很多假想对象: 或许是 IBM 或者思杰? 或者是思科, 惠普? 亦或是微软?

目前这只是一个构想。毕竟微软对使用 SUSE Linux 和从 NOVELL 购买支持服务的微软用户进行着补贴。价值约为 3.4 亿美元。再出三倍的价格就能买下整个公司了。

10 亿美元? 对微软来说是小菜一碟。但微软如何能接受在未来几年来与越来越多的 Linux 系统去分享数据中心的份额, 他们可能想拥有自己的 Linux 产品也不是完全的无稽之谈。假设开源社区一直呼吁 NOVELL 和微软抗争到底, 你只能凭着想象去猜测 SUSE 是否会继续下去, 但是这些事情最终有尘埃落定的一天。

如果微软不收购 NOVELL, SUSE 被出售给其他人会发生什么呢? 什么地方可以让微软与 Linux

共生共存，并了解到那些想要运行开源操作系统用户的想法？2006 年微软和 NOVELL 达成协议之后发生了什么？研发力量的联合，互相协作，虚拟化兼容等等都取得进展了吗？事实上，这些可能都不得不被放弃。

毋庸置疑其他公司也有掌控 NOVELL 的权利，但是如果微软公司决定收购 NOVELL，对其他公司来说就很难有机会。这是因为 NOVELL 对微软来说可能比对其他买家更具价值，微软的实力也非常雄厚。其他公司那就是红帽了。

开源软件未来发展趋势：更少利润更多理念

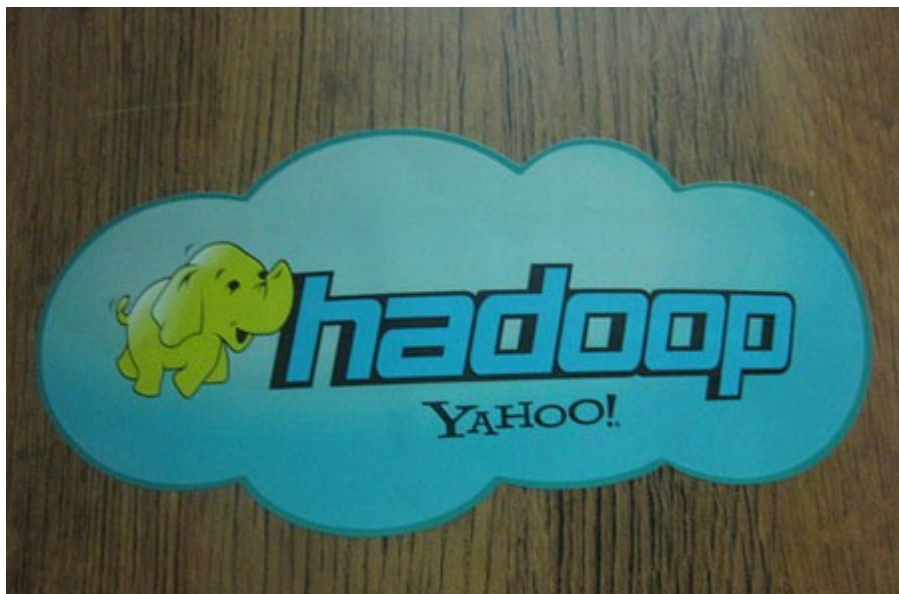
开源已经不是过去大家习惯的开源。开源是优势与不足共存的开源。

在优势方面，让我们从 19000 这个大数字开始分析。根据 Black Duck Software 的统计，19000 是从 2009 年开始执行的开源项目大概数量。正如 Black Duck 的彼得·维克索告诉我们的“尽管经济形势和 IT 预算双双吃紧，但开源社区的代码编程工作进行的依旧如火如荼”。

风险投资公司北桥创投的米歇尔·斯科认为“同时开源这个标签也不再冷漠，不再自我与众不同”。重点是开源真实的投资回报率和收益，这些对开源成为值得信赖的主流趋势会产生影响。

如果良性发展的开源软件厂商比例不断下降，过时的商业模式-即放弃代码并通过技术支持盈利的方式就不会再那么热门。斯科表示“那些处在上升阶段的厂商有更多赚钱的方法，比如红帽公司提供的延伸专业服务和咨询业务等”。

但是 Black Duck 软件公司的首席执行官提姆·伊顿认为，真正的开源爆发来自企业级应用软件周边设备。与从头开始编写代码的方式不同，企业程序开发人员可以突破公司部门界限来合作开发在开放许可证授权下共享的组件。伊顿表示“他们是有着自己开发人员参与的混合式第三方开源代码”。Black Duck 公司预测，光在美国市场上，通过使用开源代码来抵消企业级应用软件周边设备费用的市场消费能力就达到 170 亿美元。



斯科将这种开源代码的使用方式称之为“entersource”，他将其主要看做是一种合作开发的方式。斯科认为“这些项目中很少会达到事关公司成败的关键程度”。他还补充说“好的产品未必会创造出好的开源项目。事实上结果正相反，你首先需要有一个社区，然后才是社区服务的项目”。

显然，社区的灵活性是推动所有项目的首要因素。Black Duck 公司 2009 年光为 Android 就实施了 224 个新项目。不过我很意外的了解到卫生保健行业活动项目的总数量为 800 个。政府机构对开源的青睐达到了前所未有的级别，有些机构把开源看做是开放政府的基本要素。

分析成为所有开源项目中最有趣的开源领域之一。今年 1 月，InfoWorld 授予 Apache Hadoop 年度技术创新奖章，获奖理由就是 Apache Hadoop 公司创造了使用商用硬件上数千兆数据来运行大规模分析计算功能的可能性。有了 Hadoop 和开源 NoSQL 数据库来大幅度减少数据处理时间，我们就有机会带来游戏行业的变革，因为游戏程序的改动需要进行高度专业的分析。另外，昂贵的硬件和软件资源也可以供更多的专业人才使用。

有可能很多数据的处理过程都将在云上完成，云服务将提供给那些不需要全天候运行大规模计算处理的企业用户。事实上，云本身会有两种方式和开源对接。其一，多用户租赁开源软件成为缺省的软件即服务产品，其二，由谷歌、亚马逊等提供商提供的开放式应用编程接口也会被很多用户用在开源代码的研发上。虽然这还有一段适应的过程，但行业中的很多用户看似都能接受这种重新定义。

因此你可以看到：风险投资机构和开源软件厂商的底线正在动摇，如今开源正在向着几年前难以置信的层次发展-特别是当你将 IBM，甲骨文，甚至微软等主要的开源产品供应商考虑在内的时候。

收购之后 MySQL 前路多艰

甲骨文收购 Sun 之后，MySQL 的发展前景一直受到各方的密切关注。最近，一些 MySQL 长期用户向其他系统迁移的做法，为 MySQL 的未来增加了悲观的预期。

前段时间，Twitter 宣布，将淘汰既有的 MySQL 系统，改用 Cassandra 管理信息。Cassandra 是一个由 Apache 基金资助的分布式开源数据库，主要用于将海量数据分布到大量廉价服务器，进而拼凑出一个无单点故障的信息管理集群。而在 Twitter 之前，Facebook、Digg 已经开始使用 Cassandra，思科的 WebEx 也已使用 Cassandra 来收集用户反馈。

曾几何时，MySQL 作为互联网的宠儿和开源软件旗手，备受各类互联网应用的青睐。但是，随着 Sun 收购 MySQL，它曾经耀眼的光芒慢慢褪去。对许多用户而言，MySQL 已成为商业性盈利产品，其未来发展存在很大不确定性。另外，随着 Web 2.0 应用的不断扩展，很多企业发现，使用 MySQL 的成本将伴随数据量的膨胀呈指数级增长，集中式数据存储越来越难于达到效率与效益的有机平衡。

除此之外，近期兴起的 No-SQL 运动也给 IT 行业带来了新的选择。除 Cassandra 外，No-SQL 运动的代表还包括 Hadoop、Google 的 Big Table、MemCacheDB、Voldemort、CouchDB 和 MongoDB。在这样的背景下，不仅大型互联网企业开始放弃包括 MySQL 在内的关系型数据库，即便是一些企业的内部应用，考虑到多媒体、电子邮件、空间和地理信息的增多，也开始采用非关系型数据库方案。

Twitter 官方对 Cassandra 的一些评论似乎更能说明问题。Twitter 称：“Cassandra 不存在单点故障；出身于 Facebook，天生为海量数据设计；适用于大量分布式写操作；依托于一个健康的支持社区。”对于那些建立在关系型数据库之上的系统而言，Cassandra 还提供从其他关系型数据库加载数据的手段，这意味着那些潜在用户可以考虑“尝试”将其系统用于 Cassandra，而 Twitter 也正在这么做的。Twitter 计划让两套系统先并行一段时间，待确定新系统稳定运行后再将 MySQL 淘汰掉。

今天的 MySQL 不得不面临许多问题，它在大型应用领域的采用率较低，同时面临 PostgreSQL 的强有力竞争者。另外，MySQL 的草根版本正在茁壮成长，相对于官方的企业版和社区版，MySQL 的分支产品似乎得到了更多的社区支持。不仅如此，收购案并没有真正尘埃落定。尽管目前甲骨文对 Sun 的收购已经获得了美国和欧盟的认可，但能否通过我国和俄罗斯反垄断部门的审查仍需时日。

开源硬件能否像开源软件一样流行

当前开源软件的流行我们有目共睹。从嵌入式平台的 Android 到桌面平台的 Ubuntu、Firefox 和 Chromium 再到服务器平台的 LAMP，或正迅猛地吞食市场，或已打下大半江山。然而，在不为普通用户注意的角落，有一群极客在默默开发着所谓的“开源硬件”。那些“开源硬件”将来会进入大众的视线吗？

在谈开源硬件之前，让我们先来看一下开源软件为什么会流行。普通的用户绝对不会在乎一款软件是否开放源代码，使用什么许可协议。他们唯一关注的是能不能花最少的钱用上最好的软件。那么，开源软件又是怎么做到便宜和好用的呢？

- 1、免费开源软件最吸引人的地方毫无疑问是它的免费。相比于 Windows 几百上千的价格，可以自由传播，自由使用的 Ubuntu 早已向前迈了一大步。对于嵌入式系统而言，开源的系统可以降低设备制造成本，实惠消费者也是不争的事实。
- 2、高质优秀的开源项目往往能吸引大量的技术天才来协助开发。于是最终发行的软件在质量上也不会低于收费软件。就质量上 PK，开源的 Virtualbox 也没有比收费的 VMWare 逊色多少。
- 3、跨平台，作为一个 Mac 用户，我常常为找不到那些优秀的 Windows 平台软件的 Mac 版而苦恼。但是对于开源软件则没有这个问题。一个优秀的开源软件即使没有“小众操作系统”的版本，也会有爱好者自己编译。最终，优秀的开源软件常常是完美支持多平台的。例如大名鼎鼎的 AutoCAD 不原生支持 Mac，但是同样是 CAD 软件，开源的 Archimedes 却同时支持 Windows 和 Mac。所以如果我常常要在两种系统间来回切换，我显然会选择 Archimedes 来减少麻烦。
- 4、定制性强一个优秀的开源软件周围常常会有数不胜数的插件。于是软件也就会可以通过安装插件来迎合每个人的需求——如果不理解，去看看 Firefox 那恐怖的插件库吧。
- 5、问题修复及时是软件就会出问题，这是已经被无数次证实的真理。而开源软件总是可以依靠群众的智慧，在第一时间修复问题。根据 Ed bott 的博客文章，Ubuntu 和 Fedora 的更新发布频率都远高于 Windows。

什么是开源硬件

开源硬件，指与自由和开放源码软件相同方式设计的计算机和电子硬件。开源硬件开始考虑对软件以外的领域开源，是开源文化的一部分。

这个词主要是用来反映自由释放详细信息的硬件设计，如电路图、材料清单和电路板布局数据，通常使用开源软件来驱动硬件。

共享逻辑设计连同可编程逻辑器件之重构，也是一种形式的开源硬件。

一些开源硬件项目：

- * ECB AT91 – 基于 Atmel AT91RM9200 ARM9 信息处理器 (180 MHz) 的单板计算机
- * ECB ATmega32/644 – 基于 Atmel ATmega32/644 (20 MHz) 单板计算机，网络服务器性能

和少于 100mA 的电源消耗

- * Simputer – 针对发展中国家的掌上电脑
- * Open Graphics Project 目的是设计一个开放的架构和标准的显卡。
- * OpenSPARC 是一个 Sun Microsystems 贡献了 UltraSPARC T1 和 UltraSPARC T2 multicore 处理器设计之开放源码之处理器项目。
- * OpenRISC 是一个开发者工作产生非常高性能的开源 RISC 中央处理器。
- * LEON 是一个 ESA 创建的开源 SPARC-like 的开源 32 位中央处理器。是欧洲太空业标准中央处理器。
- * OpenCores 是一个基金会，试图形成一个设计师社区，以支持开源核心（逻辑设计）之处理器、外围设备和其他设备。OpenCores 维持一个叫做 Wishbone 的开放源码之芯片上互连总线规范要求。（via 维基百科）

简而言之，开源软件开放源代码，开源硬件开放电路设计。

开源硬件流行的基础：可编程逻辑器件（PLD）

不像硬件，软件的开发，传播和部署几乎可以做到零成本。但是如果使用“一次性”的电路，开源硬件别说投入大规模使用，连开发都成问题——开发者必须忍受极其缓慢的模拟器。但是，可编程逻辑器件的存在给开源硬件带来了希望。

简单的说来，可编程逻辑器件就是一个存储器加上一个逻辑门阵列。存储器上的数据可以控制逻辑门之间线路的通断。有的可编程逻辑器件甚至可以反复刷写，多次改变内部的电路排列。

如此，开源硬件也变得如软件一般可以“零成本开发，零成本部署”了。

开源软件的优点可以移植到硬件上吗

毕竟软件和硬件是不同的。开源软件的优势在硬件上仍然存在吗？

1、价格一眼看来，似乎开源硬件会有价格上的优势，因为产品价格不包含可观的研发成本。但是硬件早已不是可以“动手制作”了。然而，订做小批量的产品是非常昂贵的。让爱好自由的极客统一起来订上大批量的开源硬件又是几乎不可能的事。因此唯一可行的把开源硬件设计“落实”的方法就是把电路置于可编程逻辑器件当中。

2、品质开源软件往往是由一个庞大的开发者社区“催熟”的。但是开源硬件的开发者还不够多。但这不是个问题，毕竟开发者是可以培养的。最终，优秀的开源硬件设计是完全有可能出现的。

3、跨平台

很遗憾，硬件是无法“跨平台”的。举个大家熟悉的例子：即使是同为 x86 架构，Intel,AMD 和 VIA 的处理器还是差异巨大的。因为整个平台的不同，所谓的移植实施起来难度也大到几乎不可能实现。有人可能会提到全美达的 Efficeon，但是这样的架构在可编程逻辑器件当中即使能实现也不会有满意的性能，而且还有各平台针脚定义不同的问题。

4、定制性开源硬件可以通过制订内部总线标准去实现扩展性。但是接下来的问题是可编程逻辑器件的容量往往会极大的限制硬件的扩展性。

5、更新可编程逻辑器件的使用让零成本更新硬件成为现实。于是更新的问题并不会过多阻碍开源硬件的发展。可编程逻辑器件使硬件性能和功能失去竞争力。

前面提到，可编程逻辑器件让开源硬件的传播成为可能。但立于可编程逻辑器件之上的硬件还

可以依靠性能和功能去打动用户吗？

可编程逻辑器件的原理是由存储器内的数据决定逻辑门之间线路的通断。所以可编程逻辑器件的结构决定了它不能承载太复杂的设计，而且功耗高，速度慢。所以，可编程逻辑器件的使用不仅限制开发者的思维空间，而且拖慢速度。对于移动设备而言，可编程逻辑器件还会降低续航能力。

开源硬件流行？不太可能。

看到这里，你应该已经明白按照现有的科技和组织方式，开源硬件只会在极客当中传播。一个可能的例外是当一个开源硬件设计足够成熟的时候，可能有人会投资大规模制造。这样开源硬件则可能会大众化。

Linux 操作系统在中国市场的四大发展障碍

Linux 操作系统在国内的 OS 市场中有很不错的发展状态。但是在高一级的普及上还是存在不少的问题。下面我们就分析下有关于 Linux 操作系统在当下的市场中发展的阻碍。

Linux 操作系统不是解决中国发展过程中存在的所有问题的万能药，它只是为我们提供了一个可以明显降低成本、促进国际合作和获得更广阔发展前景的选项之一。通常情况下，雇用国外优秀管理和技术人员所需的费用会高的惊人。

不仅要向这些人支付高额的酬金，同时要为他们营造一个合适的的生活和工作环境。而他们若要全身心的为你的事业做出贡献，必须要适应中国的人文环境。这几乎不是一个中小企业或个人能够承受的。在这里，我们来研究一下那些阻碍 Linux 操作系统及开源系统在中国普及发展的重要因素。这些因素在过去的十多年甚至更长的历史时期对技术的引进产生了巨大的阻力。中国正在经历着翻天覆地的变化，这里所列出的许多因素也会随着改革的进程，随着时间而消失。

1、节约思想和利益的最大化

中国有着数千年的文化传统，在这些传统中，“节约”和“利益最大化”的影响极为深远。因为中国长期处于农业社会，生产模式局限于“有限资源”和“有限产出”。人们考虑最多的是：怎样利用手中少量的资金和土地来满足基本的衣食温饱？考虑的范围也仅仅是围绕着自家的一亩三分地。在生产环节缺乏一个工业流程式的整体规划。先撇开版权不谈，仅就市场上流行的软件价值和获取成本来看。随便一个刚刚入门、略知皮毛的年轻人都会在自己的计算机上安装上 Windows 系统，然后是 Office 套件、photoshop、AdobeReader、3DMAX 等业内顶尖的软件。

这些知名软件在中国的市场占有率达到了 95%以上。而获得这类软件所付出的代价是在地摊上购买一张光盘的价格，大约是 5-10 元。也许你会说：“这不对，你们不该用盗版的！正版的一套软件都至少要上千元。”是的，我所说的是我们都熟悉而又避而不谈的真实价格。哪怕你真的曾经花上千元购买过正版软件，或许就当你重新安装系统时，你会不由自主地选择了盗版。

为什么人们只肯花 10 元而不是花费 1000 元来获得一套同样价值的软件呢？请注意，我所说是“同样价值”！你确实没有证据说明正版软件比盗版的多了哪些功能，如果你真的能找出这种差别，那一定是没有发现这市场上还存在着“破解版”。你在选择软件时不由自主的选择了花费 10 元获得的软件，这不是你的错。这是由中国人性中根深蒂固的“节约”和“利益最大化”意识所驱使的——花最少的钱拥有最多的功能。不管这个过程是否合法，但至少是合情合理的。于是一套软件的内在价值和生产成本在这个市场中已经完全没有了参考价值。

软件越是精良，盗版也越是猖獗，获取成本也越是接近于零。在这样一个环境中，我们拿着“

免费”的 Linux 操作系统来进行推销已经没有了获取成本上的优势。“更加节约”无从谈起，名气也不够响亮，谁会选择你呢？

2、应用程序的可用性和成熟性

检验一个软件程序好坏的标准首先是考虑程序的可用性。如果你花费了大量资源研发出一套极为复杂的程序，而这套程序连一些基本的需求都无法满足，那就根本谈不上可用。可用性其实是一个相对概念。Linux 操作系统是由一些西方不了解中国的人构想、研发的，在这个过程中，由于合作的缺失，交流的滞后，Linux 操作系统在过去很长一段时间对于中国人来说根本不具备可用性。最关键的缺失是文字的显示和输入。前几年，当 Linux 操作系统已经风靡全球时，我们甚至还不能在 Linux 操作系统中正确显示中文。如果我们拿着这样一套随时都会出现乱码的软件向民众游说推荐，恐怕连我们自己都不敢相信。是的，在过去 10 多年中，中国也出现过许许多多的 Linux 操作系统相关的企业，这其中有多少人真的相信过自己的产品？

长期以来，我们的政府和科研部门没有为我们提供可以依赖的国家标准，哪怕是近乎标准的指导性文件、通用的数据结构、文档类型定义、可供广泛使用的开放中文字体，甚至是一个简单编辑器，没有！这种困境直到 UTF-8 编码的实施和“文泉驿”项目的成功后才得到了改观。

成熟性也是一个相对概念。成熟性的另一个表现就是“企业级应用就绪”。IT 企业或传统行业中的 IT 部门掌握了企业生存发展所必需的基础应用软件。其中包括日常办公套件、服务器、规划工具、部署工具、管理工具等关键部件。虽然近年来 Linux 操作系统在企业级应用方面已经取得了重大进步，特别是 WEB 和数据库服务器领域，Apache 和 MYSQL 的成绩已经促使一些企业开始关注开源软件，并加以部署实施。但是相对于普遍存在的廉价盗版软件所形成的商业环境，Linux 操作系统在中国还缺乏一套替代解决方案。企业在人员招募、培训和产品研发、兼容等方面需要的是更加直接有效的策略。面对 Linux 操作系统这样全新的环境，所有这些都是个难题。

3、落后而单一的教育体系

前面我们讲到，在企业中进行人员招募、培训、管理、协作的全面开放式改造会陷入困境，根本的原因在于中国落后而单一的教育体系。教师素质水平的落后；教材内容取材的单一落后；内容严重依赖单一企业产品；硬件设备的落后；师资力量的分配不均；学习和实践的严重脱节；急功近利的学术风气……

很难想象，一个病态的体制下能够培养出多么优秀的人才。以一种过气的浮躁去培养新的浮躁，如此恶性循环几轮之后，整个社会风气也随之败坏。所以有人以盗版为荣了，有人为蝇头小利而沾沾自喜，有些人麻木，有些人干脆死心。在这个时刻，我们说要引入如 Linux 操作系统等先进优秀的协作模式、生活方式来改善我们自己的环境几乎是天方夜谭。这个落后的体系总有一天要被人所唾弃，然后是变革。这就像当初的中国武术流派一样，拒绝对外公开、相互切磋、固守着老把式，须要有李小龙这样的变革者来冲击，须要有全新的理论来引导，然后才能前进。

4、狭隘的民族意识和国家安全

每当谈及软件、垄断、盗版问题时，总有那么些所谓高官、行业领袖、专家站出来呼吁“研发自主知识产权的操作系统，要发展民族软件……才会对国家安全有保障……”可是到底什么才是“自主知识产权”？什么才是“民族产业”？什么才是“国家安全”？研发所谓的民族软件难道就可以不用国外已经成熟的开发语言了吗？难道就可以完全另造一套硬件体系了吗？有了完全孤立开发的程序，国家安全就有谱了？答案是否定的！

就像密码学一样，世界上没有一种加密方式是可以依靠密而不宣就得到安全性的。一旦代码、

产品被生产出来，便迟早要公诸于世，软件代码也如此。你不能指望依靠“保密条款”或“版权条款”来获得安全。任何一种难以破解的秘密系统安全性是通过其内在优良设计来实现的，而不是人为意识层面的私密版权。

目前中国的困境是依赖着少数几个企业的产品，依赖着他们的私有版权。而那些高官、专家的建议无非是让我们创造一种新的“少数”去替换现有的“少数”，用另一种“私有”去替换现有的“私有”，用另一种垄断去替换现有的垄断。说到底，还是用了一种新的“不自由”去替代了现有的“不自由”。所谓的自主产业我们也有所耳闻了。其中除了为导弹、卫星、火箭等航天科技提供的自主软件还可以让人信赖一下以外，我们还没有见过哪个自主软件能够和国外产品真刀真枪抢夺市场的。剩下活着的几个，要么就是靠政府关系养着，要么就是靠坑蒙拐骗还忽悠着。社区中自发组织研发的对国外发行版的改进版也不少，不过这些在政府眼中终究是草寇游勇不足挂齿。

联想的软肋是开源软件？

3月23日，我在西安采访了IBM“五常委”之一的全球高级副总裁兼软件集团总经理 Steve Mills 先生。虽然是个纯软件的策略发布会，仍然有记者问 Steve Mills，当初 IBM 为什么要卖掉 PC 业务。Steve Mills 的回答倒是非常干脆：“我们在 PC 业务上没有竞争力”。对于 IBM 这头大象来说，它更专注能够带来高附加值的业务，在卖掉 PC 硬件之后，Steve Mills 仍然领导 IBM 软件集团做了多次关于软件的收购，将 IBM 的软件收入从 80 亿美元发展到了 220 亿美元的规模，给 IBM 贡献了 40% 以上的利润。

2004 年年底，联想集团以“蛇吞象”的大无畏气概收购了 IBM 的 PC 业务。在收购之前，联想已经是国内和亚太的 PC 老大。收购之后，联想开始成为全球 PC 行业“有竞争力”的玩家。从收购到现在，联想一直都是全球 PC 四强之一。

能够成为全球有竞争力的企业，联想的功力不可谓不深。不过，如果我们仔细分析一下联想的核心竞争力，我们还是感觉到有一丝担心。

大家都知道，联想一开始是做汉卡起家的，走的是技术路线。不过后来是通过代理 AST 等国外厂商的 PC 销售做大做强的，开始走典型的“贸工技”路线。多年在市场上的拼搏，使得联想一直非常精通渠道管理和品牌建设。联想在中国建立的“大联想”渠道体系可谓坚不可摧，无论哪家 PC 厂商进攻中国市场，渠道都将成为他们无法逾越的障碍，因为最有价值的渠道基本上都在联想手里了。当然，在品牌建设上联想也可圈可点，特别是通过赞助奥运会，进一步提升了联想公司和产品的整体美誉度。

不过，虽然身处强调技术创新的 IT 领域，联想却一直没有给外界留下太多技术创新的印象。当然，在过去微软和英特尔(博客)垄断 PC 行业的年代，产品同质化非常严重，技术创新也并不是 PC 品牌厂商们最重要的竞争力；相对而言，规模可能更为重要，因为能够从两家垄断供应商那里争取到更好的资源和价格。

也正是由于技术创新的不足，联想差点错过了从台式机向笔记本电脑升级的大潮。2003 年，从销量上计算，已经是国内 PC 老大的联想竟然在笔记本产品上输给了当时的 IBM PCD 部门。当时，联想自主设计的笔记本，无论是外形还是功能，均不敢恭维。

仅仅从这一点来看，联想收购 IBM PC 业务就是值得的。收购之后，联想在笔记本的设计能力大大地提升了一步，如今自主设计的 IdeaPad 已经丝毫不亚于惠普、宏碁、戴尔等老牌竞争对手，ThinkPad X300 更是给人惊艳的感觉。

不过，如今行业的形势又发生了变化，只有硬件的设计和创新的能力已经不够了。3C 融合、云计算、SaaS、移动互联网这些新东西都在冲击着已经有些老态龙钟的 PC 行业。未来，仅仅依靠硬件已经赚不了多少钱了，品牌厂商需要销售的是软件一体化的产品，还要加上好的服务模式。当时，这些都需要软件能力的支持。

苹果公司就是个典型的例子。没有人说苹果是家技术型公司，但实际上苹果的软件能力非常强悍。举个例子，当年微软推出 Windows 操作系统的时候曾经引来一片赞誉之声，而实际上苹果早在几年前就已经有了自己的视窗操作系统。这也是为什么现在苹果又能起来的根本原因，因为苹果有自己的软件，它能够通过这些软件为消费者提供良好的产品体验和服务。

在发布新产品的时候，外界包括联想内部经常会有意无意把自己的“宝贝”跟苹果的产品做个比较。我个人觉得两者还是不太好比较的，因为苹果卖的不仅仅是硬件，还有自己独有的软件和服务。

这也是联想的软肋所在。过去，我参加了联想不少次的发布会，每次的效果都弄得很炫、很精彩。在参观产品的时候，联想的工程师也会详细地做介绍。但是，他们谈到硬件配置的时候都是头头是道，说起 PC 服务也没有问题，但是如果深究到软件层面往往就支支吾吾了，因为联想 PC 的软件大部分都是 OEM 合作伙伴的产品，自己的东西还是不够多。我们经常会看到联想推出非常好的硬件产品和服务，不过细究起来，软件部分基本上还是外包的。

有几件事情说明联想的软件能力还有待提高：去年炒了一年为中国移动定制开发的 Ophone 手机 O1，结果直到去年年底才正式发布。O1 发布之前出了很多个测试版本，却一直无法稳定下来，我觉得主要还是软件开发的问题。

有人会说，作为硬件厂商，要那么强的软件能力有什么用，只要有好的软件项目经理，能够管理和控制软件的开发进度就好了。我要说，这可能还是不够的。

今年 1 月初在美国拉斯维加斯的 CES 上，联想发布了 IdeaPad U1、Skylight 和乐 Phone 等耳目一新的产品。虽然产品已经发布，却迟迟没有推向市场，估计还在调试软件。尤其是 Skylight 和乐 Phone，都是采用与运营商合作，甚至与数据服务捆绑销售的方式，没有定制软件怎么可能卖出去？如果软件完全外包开发，进度怎么能够掌握在联想自己手中？

据我了解，由于乐 Phone 采取了很多与其他 Android 手机不同的硬件配置，致使很多基于 Android Market 的软件并不能够直接下载使用，而是需要进行二次开发。联想这么做其实也是在学苹果，通过独特的硬件配置和应用软件，形成与其他 Android 的差异化。不过，如果自己没有很强的软件开发团队，那么多种软件都需要协调外部技术团队的话，开发和验证工作何时能够完成？据说 4 月份乐 Phone 就将正式上市，从 O1 的开发过程来看，我感觉没有那么乐观。

其实，联想软件上的软肋并不是孤立的问题，它集中反映了整个中国制造业的问题。中国制造业如果要升级，微笑曲线的两端都要升级，一个是升级品牌运作能力，另一个就是升级技术能力，尤其是软件能力。

那么，联想如何才能提升“软”实力？我想还是要一步步慢慢来。首先，在联想的“创新三角”的研究院体系中，可以组建针对各种软件技术的虚拟研究团队。联想可以把自己在美国罗利的软件架构能力、日本大和的硬件适配能力和中国的软件开发能力结合起来，好好研究一下软件对于未来 PC 的影响。例如，Linux 是否会成为 PC 的主流操作系统之一？如果能够成为主流，联想应该选择哪个 Linux 版本，如何做定制开发？再比如，桌面虚拟化是否会成为潮流？联想能够把握住桌面虚拟化软件的哪几个关键点？还有，云计算将如何影响 PC 的使用？PC 厂商需要具备哪些云计算的软

件能力？据我了解，联想研究院确实也在做一些方向性的研究工作。

研究出方向以后，如何通过具体的软件开发来实现？这个可能也是联想亟待解决的问题，毕竟联想自己缺乏一支过硬的软件开发团队，也欠缺大规模软件开发的组织管理能力。我认为，联想可以在为自己服务的软件开发商中，选择开发能力较强的进行收购。收购完成之后可以先让其保持独立运作，然后渐渐将其整合到联想的产品开发团队之中。

当然，对于联想的高管们来说，除了懂得供应链、渠道销售、物料采购这些“硬”知识之外，掌握一些软件知识也是很有必要的。不管怎么说，只有当联想具备了完整的软件能力之后，自己的命运才能够真正由自己掌握。

技术沙龙

2010 数据库技术大会精彩回眸

2010 年 4 月 2-3 两天，正值清明节，由 IT168(ChinaUnix、ITPUB、IXPUB) 主办的 2010 数据库技术大会胜利落幕。这是一次代表着当前国内最高的数据库技术水平的大会，是国内各行各业 DBA 的交流平台。

云集各行各地的数据库技术高手

由于酒店预定的原因，我们被迫将会议时间定在了清明节期间，开始，有不少人跟我建议，担心因为节日影响参会人群的积极性。说实在的，对于搞 DBA 的人来说，节日消减不了他们的积极性。

事实上，本次大会实际数据库相关参与者达到了 550 人，加上庞大的出版社、赞助商等参与者，大会总人数接近 700 人。其中，本次大会，数据库专家（演讲专家、CTO、首席 DAB、知名数据库图书作者等）超过了 150 位，而其他参会者也是数据库领域的资深骨干工程师。

大会的外地参会人数比例接近 1/3。甚至，还有远道来自我国台湾省的几位技术朋友。4.2 号的晚宴真应该搞起来，这样，就可以请来自我国台湾的朋友们，讲讲那边的技术工作环境。（酒店方说：就餐区是公共区域，不能搞带话筒的晚宴）

这次大会，参与企业更加丰富化，参会者的技术水平都很高，交流的技术问题也很深，例如：

来自银行、证券、保险、电信等行业：

中国金融总公司、招商银行总行、交通总行、民生银行总行、徽商银行、联银通、中国电信、中国联通、东航金融、深银信投资集团、大唐移动、中国太平保险集团、信达财险、华夏人寿保险、亿阳信通、深圳证券、新疆**、歌华有线、财富软件、大连期货交易所等。

来自政府、电视、交通、医疗、教育等行业：

ATA 国际航空运输协会、河北省财政厅信息中心、湖南郴州市第一人民医院、河南工业大学、江西省邮政信息技术局、深圳邮政信息技术局、河南校信通、煤炭科学研究总院、中科院、北京大学数学科学学院、国家互联网应急中心、央视、北京邮电大学、北京联合大学、中国国际图书贸易总公司、中国农业大学、华世纪教育、中国民航、湖南卫视等。

来自制造业、物流行业等：

浙江海虹药通、北京联东集团、天士力集团、通用电气、中国数码、建发集团、SK C&C、北京联东集团上海天马微、浙江永泰集团、京华时报、经纬恒润、新比士康、中山市欧普照明股份等

来自互联网行业（电子商务、网游、门户等）

阿里集团（阿里巴巴、淘宝、支付宝、Yahoo）、腾讯、新浪、搜狐 SOHU、盛大、金山、百度、TOM、搜房、EBAY、51JOB、智联招聘、北京 外企人才网络、英才网、中国万网、中国软件、好孩子集团、酷讯、淘米网易宝支付、天涯在线、杭州斯凯网络、上海商派等。

到底哪些人在关注此会？

不用讨论，最关注此会当然是 DBA 了，呵呵。本次大会，DBA 占到了 75%。其实，我们一直

想印证一个观点，数据库技术大会，仅仅是 DBA 才关心么？

在大会准备阶段，我们将准备报名者，分成了 DBA、运维工程师、应用开发工程师这三类人群。而事实证明，我们的预想是确定的，DBA 肯定是主体。不过，还真有不少的运维工程师、应用开发工程师报名参加了本次数据库技术大会。

在大会期间，就有不少专家就表示，数据库与运维、应用开发的关系越来越紧密。例如华为的首席 DBA 李强就表示，搞应用开发的人不关心数据库，那就没法让业务应用好。

数据库是一种数据管理工具，其核心在于关键数据。而关键数据将应用开发、运维紧密地关联在了一起。这也是为何还有相当一部分运维工程师、应用开发工程师报名参会的原因。真心希望下次大会，这部分人群更多一点。

演讲专家，最值得我们感谢

最要感谢的是我们的演讲专家：陈吉平、宋一平、冯春培、盖国强、周淳、冯大辉、李天伟、张宴、李强、周卫国、黄卫、朱海峰、牛新庄、张乐奕、杨向勇、李轶楠、谭俊青、漆兴、简朝阳、宁海元、丁昊、楼方鑫、王丹敬、万正勇、卢东明、萧少聪、许超前、李朝铭等各位大师。（详细介绍可参见大会官网 <http://dtcc.it168.com/>）

这次大会，来自阿里巴巴集团的演讲专家比较多，这主要是因为，阿里巴巴集团的各个子公司，在相应领域的数据库的应用开发方面，无疑是走在国内前列。例如，陈吉平老师的分布式数据库高可用可扩展应用演讲，无疑是大会的开场重头戏。而冯春培老师的数据库主机、冯大辉老师的用户体验、楼方鑫老师的数据库性能数据分析，让我们看到了数据库的几个重要领域的展现面。

华为首席 DBA 李强老师，深入解析了数据库的架构设计和 Oracle、DB2 等几大重要数据库的选型问题，可谓荡气回肠。然而，却因为时间的限制，而提前结束了演讲，在此表示歉意。

而 Sybase 的技术总监卢东明老师的“列式数据库 + CEP = 数据分析技术的完美结合”，是一次赞助演讲（白话就是打广告），然而大家却非常关注。在卢东明老师演讲完后，在会场外，一群人围着他，还交流了近一个小时，而且人越来越多。

笔者感受很深的一句话，就是盖国强老师在最后的圆桌论坛上说的：技术分享让我们成长，分享得多，收获更多。感谢盖国强老师，因为他的这个观点，是我们的技术大会能够继续下去的理由。（当然不是指核心技术的公开）

如何才能做好用户体验？

冯大辉老师的“数据库与用户体验”的演讲题目提示我们，用户体验的重要性。而对于技术大会来说，除了演讲主题是否吸引之外，用户体验同样重要。正如上次的架构师大会，这次我们也准备了 20 个接线板，让其分布在会场过道上，虽然过道显得不雅，但保证了很多笔记本电脑的电力问题。

大会的用户体验，重要的表现就在于吃、住。4.1 日当天，酒店的住房极度紧张。此前，我们帮忙为参会者预定的房间，也并没有预定上（同时，酒店的销售经理美女还在向我抱怨：给我们的团购价太亏了）。为了给参会者解决一个标准住宿间，我们压缩了工作组房间，会务组 4 个女孩住在了一个房间，真是难为她们了。就算如此，在住宿上，我们还是存在一些不足之处，希望能够得到大家理解。

大会原计划在 4.2 日晚上进行晚宴活动，但因为酒店方不同意（公共区域不宜用话筒），而未进行。于是，临时在 4.3 日最后一个时段，增加了圆桌论坛，由冯大辉主持，有盖国强、盛大徐景春、SOHU 王丹敬、腾讯崔玉明、淘宝宁海元等人参与。圆桌论坛进行到了下午 18:20，还有近 300 人

在场，看来，大家对跟 DBA 的职业生涯问题相关的圆桌也比较关注。

这次活动中，电子工业出版社博文视点图书中心、机械工业出版社华章图书中心、人民邮电出版社、清华大学出版社、南京大学出版社为我们提供了技术书籍，在此表示感谢。（一个好消息：就在大会期间，我已经拉到了下次架构师大会的 300 本图书的赞助，估计又可以人手一本了，呵呵）

大会现场图片：



（会场座无虚席）

更多图片：<http://bbs2.chinaunix.net/thread-1686192-1-1.html>

大会相关资料下载：

大会主场 PPT：

http://linux.chinaunix.net/dtcc2010/4.2_keynote.rar

4.2 下午数据库集群备份分场：

http://linux.chinaunix.net/dtcc2010/4.2_PM_DatabaseHA.rar

4.2 下午数据库 SQL、DB2 分场：

http://linux.chinaunix.net/dtcc2010/4.2_PM_SQLServer&DB2.rar

4.3 上午开源数据库分场：

http://linux.chinaunix.net/dtcc2010/4.3_AM_OpenSourceDatabase.rar

4.3 上午 Oracle 分场：

http://linux.chinaunix.net/dtcc2010/4.3_AM_Oracle.rar

4.3 下午数据仓库分场（内存数据库）：

http://linux.chinaunix.net/dtcc2010/4.3_PM_BI.rar

4.3 下午数据库监控分场：

http://linux.chinaunix.net/dtcc2010/4.3_PM_DatabaseMonitor.rar

（SOHU 提供）Oracle EM plug-in for MySQL 的代码下载：

<http://code.google.com/p/mysql-emx/> 或 <http://sourceforge.net/projects/mysql-emx/>

技术新知

Linux 内核学习的方法论

fudan_abc 之 任桥伟

Makefile 不是 Make Love

从前在学校，混了四年，没有学到任何东西，每天就是逃课，上网，玩游戏，睡觉。毕业的时候，人家跟我说 Makefile 我完全不知，但是一说 Make Love 我就来劲了，现在想来依然觉得丢人。

毫不夸张地说，Kconfig 和 Makefile 是我们浏览内核代码时最为依仗的两个文件。基本上，Linux 内核中每一个目录下都会有一个 Kconfig 文件和一个 Makefile 文件。对于一个希望能够在 Linux 内核的汪洋代码里看到一丝曙光的人来说，将它们放在怎么重要的地位都不过分。

我们去香港，通过海关的时候，总会有免费的地图和各种指南拿，有了它们在我们手里我们才不至于无头苍蝇般迷惘的行走在陌生的街道上。即使在内地出去旅游的时候一般来说也总是会首先找份地图，当然了，这时就是要去买，拿是拿不到的，不同的地方有不同的特色，只不过有的特色是服务，有的特色是索取。

Kconfig 和 Makefile 就是 Linux Kernel 迷宫里的地图。地图引导我们去认识一个城市，而 Kconfig 和 Makefile 则可以让我们了解一个 Kernel 目录下面的结构。我们每次浏览 kernel 寻找属于自己的那一段代码时，都应该首先看看目录下的这两个文件。

利用 Kconfig 和 Makefile 寻找目标代码

就像利用地图寻找目的地一样，我们需要利用 Kconfig 和 Makefile 来寻找所要研究的目标代码。

比如我们打算研究 U 盘驱动的实现，因为 U 盘是一种 storage 设备，所以我们应该先进入到 drivers/usb/storage/ 目录。但是该目录下的文件很多，那么究竟哪些文件才是我们需要关注的？这时就有必要先去阅读 Kconfig 和 Makefile 文件。

对于 Kconfig 文件，我们可以看到下面的选项。

```
34 config USB_STORAGE_DATAFAB
35     bool "Datafab Compact Flash Reader support (EXPERIMENTAL)"
36     depends on USB_STORAGE && EXPERIMENTAL
37     help
38     Support for certain Datafab CompactFlash readers.
39     Datafab has a web page at <http://www.datafabusa.com/>.
```

显然，这个选项和我们的目的没有关系。首先它专门针对 Datafab 公司的产品，其次虽然 CompactFlash reader 是一种 flash 设备，但显然不是 U 盘。因为 drivers/usb/storage 目录下的代码是针对 usb mass storage 这一类设备，而不是针对某一种特定的设备。U 盘只是 usb mass storage 设备中的一种。再比如：

```
101 config USB_STORAGE_SDDR55
102     bool "SanDisk SDDR-55 SmartMedia support (EXPERIMENTAL)"
103     depends on USB_STORAGE && EXPERIMENTAL
```

```
104     help
105     Say Y here to include additional code to support the Sandisk SDDR-55
106     SmartMedia reader in the USB Mass Storage driver.
```

很显然这个选项是有关 SanDisk 产品的，并且针对的是 SM 卡，同样不是 U 盘，所以我们也不需要去关注。

事实上，很容易确定，只有选项 CONFIG_USB_STORAGE 才是我们真正需要关注的。

```
9 config USB_STORAGE
10     tristate "USB Mass Storage support"
11     depends on USB && SCSI
12     ---help---
13     Say Y here if you want to connect USB mass storage devices to your
14     computer's USB port. This is the driver you need for USB
15     floppy drives, USB hard disks, USB tape drives, USB CD-ROMs,
16     USB flash devices, and memory sticks, along with
17     similar devices. This driver may also be used for some cameras
18     and card readers.
19
20     This option depends on 'SCSI' support being enabled, but you
21     probably also need 'SCSI device support: SCSI disk support'
22     (BLK_DEV_SD) for most USB storage devices.
23
24     To compile this driver as a module, choose M here: the
25     module will be called usb-storage.
```

接下来阅读 Makefile 文件。

```
0 #
1 # Makefile for the USB Mass Storage device drivers.
2 #
3 # 15 Aug 2000, Christoph Hellwig
4 # Rewritten to use lists instead of if-statements.
5 #
6
7 EXTRA_CFLAGS   := -ldrivers/scsi
8
9 obj-$(CONFIG_USB_STORAGE) += usb-storage.o
10
11 usb-storage-obj-$(CONFIG_USB_STORAGE_DEBUG) += debug.o
12 usb-storage-obj-$(CONFIG_USB_STORAGE_USBAT) += shuttle_usbat.o
13 usb-storage-obj-$(CONFIG_USB_STORAGE_SDDR09) += sddr09.o
14 usb-storage-obj-$(CONFIG_USB_STORAGE_SDDR55) += sddr55.o
15 usb-storage-obj-$(CONFIG_USB_STORAGE_FREECOM) += freecom.o
16 usb-storage-obj-$(CONFIG_USB_STORAGE_DPCM) += dpcm.o
```



```
17 usb-storage-obj-$(CONFIG_USB_STORAGE_ISD200) += isd200.o
18 usb-storage-obj-$(CONFIG_USB_STORAGE_DATAFAB) += datafab.o
19 usb-storage-obj-$(CONFIG_USB_STORAGE_JUMPSHOT) += jumpshot.o
20 usb-storage-obj-$(CONFIG_USB_STORAGE_ALAUDA) += alauda.o
21 usb-storage-obj-$(CONFIG_USB_STORAGE_ONETOUCH) += onetouch.o
22 usb-storage-obj-$(CONFIG_USB_STORAGE_KARMA) += karma.o
23
24 usb-storage-objs := scsiglue.o protocol.o transport.o usb.o \
25     initializers.o $(usb-storage-obj-y)
26
27 ifneq ($(CONFIG_USB_LIBUSUAL),)
28     obj-$(CONFIG_USB) += libusual.o
29 endif
```

前面通过 Kconfig 文件的分析，我们确定了只需要去关注 CONFIG_USB_STORAGE 选项。在 Makefile 文件里查找 CONFIG_USB_STORAGE，从第 9 行得知，该选项对应的模块为 usb-storage。

因为 Kconfig 文件里的其他选项我们都不需要关注，所以 Makefile 的 11~22 行可以忽略。第 24 行意味着我们只需要关注 scsiglue.c、protocol.c、transport.c、usb.c、initializers.c 以及它们同名的.h 头文件。

Kconfig 和 Makefile 很好的帮助我们定位到了所要关注的目标，就像我们到一个陌生的地方要随身携带地图，当我们学习 Linux 内核时，也要谨记寻求 Kconfig 和 Makefile 的帮助。

透过现象看本质，兽兽门无非就是一些人体艺术展示。同样往本质里看过去，学习内核，就是学习内核的源代码，任何内核有关的书籍都是基于内核，而又不高于内核的。

既然要学习内核源码，就要经常对内核代码进行分析，而内核代码千千万，还前仆后继的不断往里加，这就让大部分人都有种雾里看花花不见的无助感。不过不要怕，孔老夫子早就留给我们了应对之策：敏于事而慎于言，就有道而正焉，可谓好学也已。这就是说，做事要踏实才是好学生好同志，要遵循严谨的态度，去理解每一段代码的实现，多问多想多记。如果抱着走马观花，得过且过的态度，结果极有可能就是一边看一边丢，没有多大的收获。

假设全国房价上涨 1.5%，假设 80 后局长是农民子弟，……，既然我们的人生充满了假设，那么我在这假设你马上就迫不及待的希望研究内核中 USB 子系统的实现，应该没有意见吧？那好，下面就以 USB 子系统的实现分析为标本看看分析内核源码应该如何入手。

分析 README

内核中 USB 子系统的代码位于目录 drivers/usb，这个结论并不需要假设。于是我们进入到该目录，执行命令 ls，结果显示如下：

```
atm class core gadget host image misc mon serial storage Kconfig
Makefile README usb-skeleton.c
```

目录 drivers/usb 共包含有 10 个子目录和 4 个文件，usb-skeleton.c 是一个简单的 USB driver 的框架，感兴趣的可以去看看，目前来说，它还吸引不了我们的眼球。那么首先应该关注什么？如果迎面走来一个 ppm，你会首先看脸、脚还是其它？当然答案依据每个人的癖好会有所不同。不过这

里的问题应该只有一个答案，那就是 Kconfig、Makefile、README。

README 里有关于这个目录下内容的一般性描述，它不是关键，只是帮助你了解。再说了，面对“read 我吧 read 我吧”这么热情奔放的呼唤，善良的我们是不可能无动于衷的，所以先来看看里面都有些什么内容。

23 Here is a list of what each subdirectory here is, and what is contained in
24 them.

25

26 core/ - This is for the core USB host code, including the
27 usbfs files and the hub class driver ("khubd").

28

29 host/ - This is for USB host controller drivers. This
30 includes UHCI, OHCI, EHCI, and others that might
31 be used with more specialized "embedded" systems.

32

33 gadget/ - This is for USB peripheral controller drivers and
34 the various gadget drivers which talk to them.

35

36

37 Individual USB driver directories. A new driver should be added to the
38 first subdirectory in the list below that it fits into.

39

40 image/ - This is for still image drivers, like scanners or
41 digital cameras.

42 input/ - This is for any driver that uses the input subsystem,
43 like keyboard, mice, touchscreens, tablets, etc.

44 media/ - This is for multimedia drivers, like video cameras,
45 radios, and any other drivers that talk to the v4l
46 subsystem.

47 net/ - This is for network drivers.

48 serial/ - This is for USB to serial drivers.

49 storage/ - This is for USB mass-storage drivers.

50 class/ - This is for all USB device drivers that do not fit
51 into any of the above categories, and work for a range
52 of USB Class specified devices.

53 misc/ - This is for all USB device drivers that do not fit
54 into any of the above categories.

这个 README 文件描述了前边使用 ls 命令列出的那 10 个文件夹的用途。那么什么是 USB Core？Linux 内核开发者们，专门写了一些代码，负责实现一些核心的功能，为别的设备驱动程序提供服务，比如申请内存，比如实现一些所有的设备都会需要的公共的函数，并美其名曰 USB Core。

时代总在发展，当年胖杨贵妃照样迷死唐明皇，而如今人们欣赏的则是林志玲这样的魔鬼身材。同样，早期的 Linux 内核，其结构并不是如今天这般有层次感，远不像今天这般错落有致，那时候

drivers/usb/这个目录下边放了很多很多文件，USB Core 与其他各种设备的驱动程序的代码都堆砌在这里，后来，怎奈世间万千的变幻，总爱把有情的人分两端。于是在 drivers/usb/目录下面出来了一个 core 目录，就专门放一些核心的代码，比如初始化整个 USB 系统，初始化 Root Hub，初始化主机控制器的代码，再后来甚至把主机控制器相关的代码也单独建了一个目录，叫 host 目录，这是因为 USB 主机控制器随着时代的发展，也开始有了好几种，不再像刚开始那样只有一种，所以呢，设计者们把一些主机控制器公共的代码仍然留在 core 目录下，而一些各主机控制器单独的代码则移到 host 目录下面让负责各种主机控制器的人去维护。

那么 USB gadget 那？gadget 说白了就是配件的意思，主要就是一些内部运行 Linux 的嵌入式设备，比如 PDA，设备本身有 USB 设备控制器（USB Device Controller），可以将 PC，也就是我们的主机作为 master 端，将这样的设备作为 slave 端和主机通过 USB 进行通信。从主机的观点来看，主机系统的 USB 驱动程序控制插入其中的 USB 设备，而 USB gadget 的驱动程序控制外围设备如何作为一个 USB 设备和主机通信。比如，我们的嵌入式板子上支持 SD 卡，如果我們希望在将板子通过 USB 连接到 PC 之后，这个 SD 卡被模拟成 U 盘，那么就要通过 USB gadget 架构的驱动。

剩下的几个目录分门别类的放了各种 USB 设备的驱动，比如 U 盘的驱动在 storage 目录下，触摸屏和 USB 键盘鼠标的驱动在 input 目录下，等等。

我们响应了 README 的热情呼唤，它便给予了我们想要的，通过它我们了解了 USB 目录里的那些文件夹都有着什么样的角色。到现在为止，就只剩下内核的地图——Kconfig 与 Makefile 两个文件了。有地图在手，对于在内核中游荡的我们来说，是件很愉悦的事情，不过，因为我们的目的是研究内核对 USB 子系统的实现，而不是特定设备或 host controller 的驱动，所以这里的定位很明显，USB Core 就是我们需要关注的对象，那么接下来就是要对 core 目录中的内容进行定位了。

分析 Kconfig 和 Makefile

进入到 drivers/usb/core 目录，执行命令 ls，结果显示如下：

```
Kconfig Makefile buffer.c config.c devices.c devio.c driver.c
endpoint.c file.c generic.c hcd-pci.c hcd.c hcd.h hub.c hub.h
inode.c message.c notify.c otg_whitelist.h quirks.c sysfs.c urb.c
usb.c usb.h
```

然后执行 wc 命令，如下所示。

```
# wc -l /*
 148 buffer.c
 607 config.c
 706 devices.c
1677 devio.c
1569 driver.c
 357 endpoint.c
 248 file.c
 238 generic.c
1759 hcd.c
 458 hcd.h
 433 hcd-pci.c
3046 hub.c
```

```
195 hub.h
758 inode.c
144 Kconfig
21 Makefile
1732 message.c
68 notify.c
112 otg_whitelist.h
161 quirks.c
710 sysfs.c
589 urb.c
984 usb.c
160 usb.h
16880 total
```

drivers/usb/core 目录共包括 24 个文件，16880 行代码。core 不愧是 core，为大家默默的做这么多事。不过这么多文件里不一定是我们所需要的关注的，先拿咱们的地图来看看接下来该怎么走。先看看 Kconfig 文件，可以看到下面的选项。

```
15 config USB_DEVICEFS
16     bool "USB device filesystem"
17     depends on USB
18     ---help---
19     If you say Y here (and to "/proc file system support" in the "File
20     systems" section, above), you will get a file /proc/bus/usb/devices
21     which lists the devices currently connected to your USB bus or
22     busses, and for every connected device a file named
23     "/proc/bus/usb/xxx/yyy", where xxx is the bus number and yyy the
24     device number; the latter files can be used by user space programs
25     to talk directly to the device. These files are "virtual", meaning
26     they are generated on the fly and not stored on the hard drive.
27
28     You may need to mount the usbfs file system to see the files, use
29     mount -t usbfs none /proc/bus/usb
30
31     For the format of the various /proc/bus/usb/ files, please read
32     <file:Documentation/usb/proc_usb_info.txt>.
33
34     Usbfs files can't handle Access Control Lists (ACL), which are the
35     default way to grant access to USB devices for untrusted users of a
36     desktop system. The usbfs functionality is replaced by real
37     device-nodes managed by udev. These nodes live in /dev/bus/usb and
38     are used by libusb.
```

选项 USB_DEVICEFS 与 usbfs 文件系统有关。usbfs 文件系统挂载在 /proc/bus/usb 目录，显示了当前连接的所有 USB 设备及总线的各种信息，每个连接的 USB 设备在其中都会有一个对应的文件进行

描述。比如文件/proc/bus/usb/xxx/yyy, xxx 表示总线的 序号, yyy 表示设备所在总线的地址。不过不能够依赖它们来稳定地访问设备, 因为同一设备两次连接对应的描述文件可能会不同, 比如, 第一次连接一个设备 时, 它可能是 002/027, 一段时间后再次连接, 它可能就已经改变为 002/048。

就好比好不容易你暗恋的 mm 今天见你的时候对你抛了个媚眼, 你心花怒放, 赶快去买了 100 块彩票庆祝, 到第二天再见到她的时候, 她对你说你是谁啊, 你悲痛欲绝的刮开那 100 块彩票, 上面清一色的谢谢你。

因为 usbfs 文件系统并不属于 USB 子系统实现的核心部分, 与之相关的代码我们可以不必关注。

```
74 config USB_SUSPEND
75     bool "USB selective suspend/resume and wakeup (EXPERIMENTAL)"
76     depends on USB && PM && EXPERIMENTAL
77     help
78     If you say Y here, you can use driver calls or the sysfs
79     "power/state" file to suspend or resume individual USB
80     peripherals.
81
82     Also, USB "remote wakeup" signaling is supported, whereby some
83     USB devices (like keyboards and network adapters) can wake up
84     their parent hub. That wakeup cascades up the USB tree, and
85     could wake the system from states like suspend-to-RAM.
86
87     If you are unsure about this, say N here.
```

这一项是有关 USB 设备的挂起和恢复。开发 USB 的人都是节电节能的好孩子, 所以协议里就规定了, 所有的设备都必须支持挂起状态, 就是说为了达到节 电的目的, 当设备在指定的时间内, 如果没有发生总线传输, 就要进入挂起状态。当它收到一个 non-idle 的信号时, 就会被唤醒。节约用电从 USB 做起。不过这个与主题也没太大关系, 相关代码也可以不用关注了。

剩下的还有几项, 不过似乎与咱们关系也不大, 还是去看看 Makefile。

```
5 usbcore-objs := usb.o hub.o hcd.o urb.o message.o driver.o \
6               config.o file.o buffer.o sysfs.o endpoint.o \
7               devio.o notify.o generic.o quirks.o
8
9 ifeq ($(CONFIG_PCI),y)
10     usbcore-objs += hcd-pci.o
11 endif
12
13 ifeq ($(CONFIG_USB_DEVICEFS),y)
14     usbcore-objs += inode.o devices.o
15 endif
16
17 obj-$(CONFIG_USB) += usbcore.o
18
```

```
19 ifeq ($(CONFIG_USB_DEBUG),y)
20 EXTRA_CFLAGS += -DDEBUG
21 endif
```

Makefile 可比 Kconfig 简略多了，所以看起来也更亲切点，咱们总是拿的 money 越多越好，看的代码越少越好。这里之所以会出现 CONFIG_PCI，是因为通常 USB 的 Root Hub 包含在一个 PCI 设备中。hcd-pci 和 hcd 顾名思义就知道是说主机控制器的，它们实现了主机控制器公共部分，按协议里的说法它们就是 HCDI（HCD 的公共接口），host 目录下则实现了各种不同的主机控制器。

CONFIG_USB_DEVICEFS 前面的 Kconfig 文件里也见到了，关于 usbfs 的，与咱们的主题无关，inode.c 和 devices.c 两个文件也可以不用管了。

那么我们可以得出结论，为了理解内核对 USB 子系统的实现，我们需要研究 buffer.c、config.c、driver.c、endpoint.c、file.c、generic.c、hcd.c、hcd.h、hub.c、message.c、notify.c、otg_whitelist.h、quirks.c、sysfs.c、urb.c 和 usb.c 文件。这么看来，好像大都需要关注的样子，没有减轻多少压力，不过这里本身就是 USB Core 部分，是要做很多的事为咱们分忧的，所以多点也是可以理解的。

下面的分析，米卢教练说了，内容不重要，重要的是态度。就像韩局长对待日记的态度那样，严谨而细致。

只要你使用这样的态度开始分析内核，那么无论你选择内核的哪个部分作为切入点，比如 USB，比如进程管理，在花费相对不算很多的时间之后，你就会发现你对内核的理解会上升到另外一个高度，一个抱着情景分析，抱着 0.1 内核完全注释，抱着各种各样的内核书籍翻来覆去的看很多遍又忘很多遍都无法达到的高度。请相信我！

让我们在 Linux 社区里发出号召：学习内核源码，从学习韩局长开始！

态度决定一切：从初始化函数开始

任小强们说房价高涨从现在开始，股评家们说牛市从 5000 点开始。他们的开始需要我们的钱袋，我们的开始只需要一台电脑，最好再有一杯茶，伴着几支小曲儿，不盯着钱总是会比较惬意的。生容易，活容易，生活不容易，因为总要盯着钱。

有了地图 Kconfig 和 Makefile，我们可以在庞大复杂的内核代码中定位以及缩小了目标代码的范围。那么现在，为了研究内核对 USB 子系统的实现，我们还需要在目标代码中找到一个突破口，这个突破口就是 USB 子系统的初始化代码。

针对某个子系统或某个驱动，内核使用 subsys_initcall 或 module_init 宏指定初始化函数。在 drivers/usb/core/usb.c 文件中，我们可以发现下面的代码。

```
940 subsys_initcall(usb_init);
941 module_exit(usb_exit);
```

我们看到一个 subsys_initcall，它也是一个宏，我们可以把它理解为 module_init，只不过因为这部分代码比较核心，开发者们把它看作一个子系统，而不仅仅是一个模块。这也很好理解，usbcore 这个模块它代表的不是某一个设备，而是所有 USB 设备赖以生存模块，Linux 中，像这样一个类别的设备驱动被归结为一个子系统。比如 PCI 子系统，比如 SCSI 子系统，基本上，drivers/目录下面第一层的每个目录都算一个子系统，因为它们代表了一类设备。

subsys_initcall(usb_init)的意思就是告诉我们 usb_init 是 USB 子系统真正的初始化函数，而

usb_exit() 将是整个 USB 子系统的结束时的清理函数。于是为了研究 USB 子系统在内核中的实现，我们需要从usb_init 函数开始看起。

```
865 static int __init usb_init(void)
866 {
867     int retval;
868     if (nousb) {
869         pr_info("%s: USB support disabled\n", usbcore_name);
870         return 0;
871     }
872
873     retval = ksuspend_usb_init();
874     if (retval)
875         goto out;
876     retval = bus_register(&usb_bus_type);
877     if (retval)
878         goto bus_register_failed;
879     retval = usb_host_init();
880     if (retval)
881         goto host_init_failed;
882     retval = usb_major_init();
883     if (retval)
884         goto major_init_failed;
885     retval = usb_register(&usbfs_driver);
886     if (retval)
887         goto driver_register_failed;
888     retval = usb_devio_init();
889     if (retval)
890         goto usb_devio_init_failed;
891     retval = usbfs_init();
892     if (retval)
893         goto fs_init_failed;
894     retval = usb_hub_init();
895     if (retval)
896         goto hub_init_failed;
897     retval = usb_register_device_driver(&usb_generic_driver, THIS_MODULE);
898     if (!retval)
899         goto out;
900
901     usb_hub_cleanup();
902 hub_init_failed:
903     usbfs_cleanup();
904 fs_init_failed:
905     usb_devio_cleanup();
```

```
906 usb_devio_init_failed:
907  usb_deregister(&usbfs_driver);
908 driver_register_failed:
909  usb_major_cleanup();
910 major_init_failed:
911  usb_host_cleanup();
912 host_init_failed:
913  bus_unregister(&usb_bus_type);
914 bus_register_failed:
915  ksuspend_usb_cleanup();
916 out:
917  return retval;
918 }
```

(1) __init 标记。

关于 usb_init，第一个问题是，第 865 行的__init 标记具有什么意义？

写过驱动的应该不会陌生，它对内核来说就是一种暗示，表明这个函数仅在初始化期间使用，在模块被装载之后，它占用的资源就会释放掉用作它处。它的暗示你懂，可你的暗示，她却不懂或者懂装不懂，多么让人感伤。它在自己短暂的一生中一直从事繁重的工作，吃的是草吐出的是牛奶，留下的是整个 USB 子系统的繁荣。

受这种精神所感染，我觉得有必要为它说的更多些。__init 的定义在 include/linux/init.h 文件里

```
43 #define __init      __attribute__((__section__ (".init.text")))
```

好像这里引出了更多的疑问，__attribute__是什么？Linux 内核代码使用了大量的 GNU C 扩展，以至于 GNU C 成为能够编译内核的唯一编译器，GNU C 的这些扩展对代码优化、目标代码布局、安全检查等方面也提供了很强的支持。而__attribute__就是这些扩展中的一个，它主要被用来声明一些特殊的属性，这些属性主要被用来指示编译器进行特定方面的优化和更仔细的代码检查。GNU C 支持十几个属性，section 是其中的一个，我们查看 GCC 的手册可以看到下面的描述

`'section ("section-name")'`

Normally, the compiler places the code it generates in the `'text'` section. Sometimes, however, you need additional sections, or you need certain particular functions to appear in special sections.

The `'section'` attribute specifies that a function lives in a particular section. For example, the declaration:

```
extern void foobar (void) __attribute__((section ("bar")));
```

puts the function `'foobar'` in the `'bar'` section.

Some file formats do not support arbitrary sections so the `'section'` attribute is not available on all platforms. If you need to map the entire contents of a module to a particular section, consider using the facilities of the linker instead.

通常编译器将函数放在.text 节，变量放在.data 或.bss 节，使用 section 属性，可以让编译器将函

数或变量放在指定的节中。那么前面对__init的定义便表示将它修饰的代码放在.init.text节。连接器可以把相同节的代码或数据安排在一起,比如__init修饰的所有代码都会被放在.init.text节里,初始化结束后就可以释放这部分内存。

问题可以到此为止,也可以更深入,即内核又是如何调用到这些__init修饰的初始化函数?要回答这个问题,还需要回顾一下subsys_initcall宏,它也在include/linux/init.h里定义

```
125 #define subsys_initcall(fn)      __define_initcall("4",fn,4)
```

这里又出现了一个宏__define_initcall,它用于将指定的函数指针fn放到initcall.init节里而对于具体的subsys_initcall宏,则是把fn放到.initcall.init的子节.initcall4.init里。要弄清楚.initcall.init、.init.text和.initcall4.init这样的东东,我们还需要了解一点内核可执行文件相关的概念。

内核可执行文件由许多链接在一起的对象文件组成。对象文件有许多节,如文本、数据、init数据、bss等等。这些对象文件都是由一个称为链接器脚本的文件链接并装入的。这个链接器脚本的功能是将输入对象文件的各节映射到输出文件中;换句话说,它将所有输入对象文件都链接到单一的可执行文件中,将该可执行文件的各节装入到指定地址处。vmlinux.lds是存在于arch/<target>/目录中的内核链接器脚本,它负责链接内核的各个节并将它们装入内存中特定偏移量处。

我可以负责的告诉你,要看懂vmlinux.lds这个文件是需要一番功夫的,不过大家都是聪明人,聪明人做聪明事,所以你需要做的只是搜索initcall.init,然后便会看到似曾相识的内容

```
__initcall_start = .;
.initcall.init : AT(ADDR(.initcall.init) - 0xC0000000) {
*(.initcall1.init)
*(.initcall2.init)
*(.initcall3.init)
*(.initcall4.init)
*(.initcall5.init)
*(.initcall6.init)
*(.initcall7.init)
}
__initcall_end = .;
```

这里的__initcall_start指向.initcall.init节的开始,__initcall_end指向它的结尾。而.initcall.init节又被分为了7个子节,分别是

```
.initcall1.init
.initcall2.init
.initcall3.init
.initcall4.init
.initcall5.init
.initcall6.init
.initcall7.init
```

我们的subsys_initcall宏便是将指定的函数指针放在了.initcall4.init子节。其它的比如core_initcall将函数指针放在.initcall1.init子节,device_initcall将函数指针放在了.initcall6.init子节等等,都

可以从 include/linux/init.h 文件找到它们的定义。各个字节的顺序是确定的，即先调用 .initcall1.init 中的函数指针再调用 .initcall2.init 中的函数指针，等等。__init 修饰的初始化函数在内核初始化过程中调用的顺序和 .initcall.init 节里函数指针的顺序有关，不同的初始化函数被放在不同的子节中，因此也就决定了它们的调用顺序。

至于实际执行函数调用的地方，就在 /init/main.c 文件里，内核的初始化么，不在那里还能在哪里，里面的 do_initcalls 函数会直接用到这里的 __initcall_start、__initcall_end 来进行判断。

(2) 模块参数。

关于 usb_init 函数，第二个问题是，第 868 行的 nousb 表示什么？

知道 C 语言的人都会知道 nousb 是一个标志，只是不同的标志有不一样的精彩，这里的 nousb 是用来让我们在启动内核的时候通过内核参数去掉 USB 子系统的，Linux 社会是一个很人性化的世界，它不会去逼迫我们接受 USB，一切都只关乎我们自己的需要。不过我想我们一般来说是不会去指定 nousb 的吧。如果你真的指定了 nousb，那它就只会幽怨的说一句 “USB support disabled”，然后退出 usb_init。

nousb 在 drivers/usb/core/usb.c 文件中定义为：

```
static int nousb; /* Disable USB when built into kernel image */
module_param_named(autosuspend, usb_autosuspend_delay, int, 0644);
MODULE_PARM_DESC(autosuspend, "default autosuspend delay");
```

从中可知 nousb 是个模块参数。关于模块参数，我们都知道可以在加载模块的时候可以指定，但是如何在内核启动的时候指定？

打开系统的 grub 文件，然后找到 kernel 行，比如：

```
kernel /boot/vmlinuz-2.6.18-kdb root=/dev/sda1 ro splash=silent vga=0x314
```

其中的 root, splash, vga 等都表示内核参数。当某一模块被编译进内核的时候，它的模块参数便需要在 kernel 行来指定，格式为 “模块名.参数=值”，比如：

```
modprobe usbcore autosuspend=2
```

对应到 kernel 行，即为：

```
usbcore.autosuspend=2
```

通过命令 “modinfo -p \${modulename}” 可以得知一个模块有哪些参数可以使用。同时，对于已经加载到内核里的模块，它们的模块参数会列举在 /sys/module/\${modulename}/parameters/ 目录下面，可以使用 “echo -n \${value} > /sys/module/\${modulename}/parameters/\${parm}” 这样的命令去修改。

(3) 可变参数宏。

关于 usb_init 函数，第三个问题是，pr_info 如何实现与使用？

pr_info 只是一个打印信息的可辨参数宏，printk 的变体，在 include/linux/kernel.h 里定义：

```
242 #define pr_info(fmt,arg...) \
243     printk(KERN_INFO fmt,##arg)
```

99 年的 ISO C 标准里规定了可变参数宏，和函数语法类似，比如

```
#define debug(format, ...) fprintf (stderr, format, __VA_ARGS__)
```

里面的“...”就表示可变参数，调用时，它们就会替代宏体里的__VA_ARGS__。GCC 总是会显得特立独行一些，它支持更复杂的形式，可以给可变参数取个名字，比如

```
#define debug(format, args...) fprintf (stderr, format, args)
```

有了名字总是会容易交流一些。是不是与 pr_info 比较接近了？除了‘##’，它主要是针对空参数的情况。既然说是可变参数，那传递空参数也总是可以的，空即是多，多即是空，股市里的哲理这里同样也是适合的。如果没有‘##’，传递空参数的时候，比如

```
debug ("A message");
```

展开后，里面的字符串后面会多个多余的逗号。这个逗号你应该不会喜欢，而‘##’则会使预处理器去掉这个多余的逗号。

关于 usb_init 函数，上面的三个问题之外，余下的代码分别完成 usb 各部分的初始化，接下来就需要围绕它们分别进行深入分析。因为这里只是演示如何入手分析，展示的只是一种态度，所以具体的深入分析就免了吧。

对于学习来说，无论是在学校的课堂学习，还是这里说的内核学习，效果好或者坏，最主要取决于两个方面——方法论和心理。注意，我无视了智商的差异，这玩意儿玄之又玄，岔开了说，属于迷信的范畴。

前面又是 Kernel 地图，又是如何入手，说的都是方法论的问题，那么这里要面对的就主要是心理上的问题。

而心理上的问题主要有两个，一个是盲目，就是在能够熟练适用 Linux 之前，对 Linux 为何物还说不出道道来，就迫不及待的盲目的去研究内核的源代码。这一部分人会觉得既然是学习内核，那么耗费时间在熟悉 Linux 的基本操作上纯粹是浪费宝贵的时间和感情。不过这样虽然很有韩峰同志的热情和干劲儿，但明显走入了一种心理误区。重述 Linus 的那句话：要先会使用它。

第二个就是恐惧。人类进化这么多年，面对复杂的物体和事情还是总会有天生的惧怕感，体现在内核学习上面就是：那么庞大复杂的内核代码，让人面对起来该情何以堪啊！

有了这种恐惧无力感存在，心理上就会去排斥面对接触内核源码，宁愿去抱着情景分析，搜集各种各样五花八门的内核书籍放在那里屯着，看了又忘，忘了又看，也不大情愿去认真细致得浏览源码。

这个时候，我们在心理上是脆弱得，我们忘记了芙蓉姐姐，工行女之所以红起来，不是她们有多好，而是因为她们得心理足够坚强。是的，除了向韩局长学习态度，我们还要向涌现出来的无数个芙蓉姐姐和工行女学习坚强的心理。

有必要再强调一次，学习内核，就是学习内核的源代码，任何内核有关的书籍都是基于内核，而又不高于内核的。内核源码本身就是最好的参考资料，其他任何经典或非经典的书最多只是起到个辅助作用，不能也不应该取代内核代码在我们学习过程中的主导地位。

“世界上最缺的不是金钱，而是资源。”当我在一份报纸上看到这句大大标题时，我的第一反应是——作者一定是个自然环保主义者，然后我在羞愧得反省自身的同时油然而生一股对这样的无产主义理想者无比崇敬的情绪来。

于是，我继续往下看，“因此在 XXX 还未正式面市之时，前来咨询的客户已经不少，这些有眼光的购房者明白，谁能在目前最好的购房机会下最大化地占有绝版资源，谁就掌控了未来财富流向。

”（为了避免做广告的嫌疑，请允许我使用 XXX 代替该楼盘的名字。）顿时，我悟道了！

其实，韩峰同志已经在日记里告诉了我们资源的重要性，因此我们在学习韩峰同志严谨细致的态度同时，还要领悟他对资源的灵活运用。只有在以内核源码为中心，坚持各种学习资源的长期建设不动摇，才能达到韩局长那样的高度，俯视 Linux 内核世界里的人生百态。

注意，这个观点与前面所说的学习效果主要取决于方法论和心理两个方面并不矛盾，它们属于不同层次上的问题。

内核文档

内核代码中包含有大量的文档，这些文档对于学习理解内核有着不可估量的价值，记住，在任何时候，它们在我们心目中的地位都应该高于那些各式的内核参考书。下面是一些内核新人所应该阅读的文档。

README

这个文件首先简单介绍了 Linux 内核的背景，然后描述了如何配置和编译内核，最后还告诉我们出现问题时应该怎么办。

Documentation/Changes

这个文件给出了用来编译和使用内核所需要的最小软件包列表。

Documentation/CodingStyle

这个文件描述了内核首选的编码风格，所有代码都应该遵守里面定义的规范。

Documentation/SubmittingPatches

Documentation/SubmittingDrivers

Documentation/SubmitChecklist

这三个文件都是描述如何提交代码的，其中 SubmittingPatches 给出创建和提交补丁的过程，SubmittingDrivers 描述了如何将设备驱动提交给 2.4、2.6 等不同版本的内核树，SubmitChecklist 则描述了提交代码之前需要 check 自己的代码应该遵守的某些事项。

Documentation/stable_api_nonsense.txt

这个文件解释了为什么内核没有一个稳定的内部 API（到用户空间的接口——系统调用——是稳定的），它对于理解 Linux 的开发哲学至关重要，对于将开发平台从其他操作系统转移到 Linux 的开发者来说也很重要。

Documentation/stable_kernel_rules.txt

解释了稳定版内核（stable releases）发布的规则，以及如何将补丁提交给这些版本。

Documentation/SecurityBugs

内核开发者对安全性问题非常关注，如果你认为自己发现了这样的问题，可以根据这个文件中给出的联系方式提交 bug，以便能够尽可能快的解决这个问题。

Documentation/kernel-docs.txt

这个文件列举了很多内核相关的文档和书籍，里面不乏经典之作。

Documentation/applying-patches.txt

这个文件回答了如何为内核打补丁。

Documentation/bug-hunting

这个文件是有关寻找、提交、修正 bug 的。

Documentation/HOWTO

这个文件将指导你如何成为一名内核开发者，并且学会如何同内核开发社区合作。它尽可能不包括任何关于内核编程的技术细节，但会给你指引一条获得这些知识的正确途径。

经典书籍

待到山花烂漫时，还是那些经典在微笑。

有关内核的书籍可以用汗牛充栋来形容，不过只有一些经典的神作经住了考验。首先是 5 本久经考验的神作（个人概括为“2+1+2”，第一个 2 是指 2 本全面讲解内核的书，中间的 1 指 1 本讲解驱动开发的书，后面的 2 则指 2 本有关内核具体子系统的书，你是否想到了某某广告里三个人突然站起单臂齐举高呼“1 比 1 比 1”的场景？）。

《Linux 内核设计与实现》

简称 LKD，从入门开始，介绍了诸如进程管理、系统调用、中断和中断处理程序、内核同步、时间管理、内存管理、地址空间、调试技术等方面，内容比较浅显易懂，个人认为是内核新人首先必读的书籍。新人得有此书，足矣！

《深入理解 Linux 内核》

简称 ULK，相比于 LKD 的内容不够深入、覆盖面不广，ULK 要深入全面得多。

前面这两本，一本提纲挈领，一本全面深入。

《Linux 设备驱动程序》

简称 LDD，驱动开发者都要人手一本了。

《深入理解 Linux 虚拟内存管理》

简称 LVMM，是一本介绍 Linux 虚拟内存管理机制的书。如果你希望深入的研究 Linux 的内存管理子系统，仔细的研读这本书无疑是最好的选择。

《深入理解 LINUX 网络内幕》

一本讲解网络子系统实现的书，通过这本书，我们可以了解到 Linux 内核是如何实现复杂的网络功能的。

（忘了声明下，我这列出来的书名是中文的，但是并不代表我建议大家去看他们的中文版，其中有的翻译的实在太……了，呵呵）

这 5 本书各有侧重，正如下面的图所展示的那样，恰好代表了个人一直主张的内核学习方法：首先通过 LKD 或 ULK 了解内核的设计实现特点，对内核有个整体全局的认识和理解，然后可分为两个岔路，如果从事驱动开发，则钻研 LDD，如果希望对内核不是泛泛而谈而是有更深入的理解，则可以选择一个自己感兴趣的子系统，仔细分析它的代码，不懂的地方就通过社区、邮件列表或者直接发 Email 给 maintainer 请教等途径弄懂，切勿得过且过，这样分析下来，对同步、中断等等内核的很多机制也同样会非常了解，俗话说的一通则百通就是这个道理。当然，如果你选择研究的是内存管理或者网络，则可以有上面的两本书可以学习，如果是其他子系统，可能就没有这么好的运气了。



内核社区

最近几年，社区网站非常的热火，不过此社区非彼社区。

Linux 最大的一个优势就是它有一个紧密团结了众多使用者和开发者的社区，它的目标就是提供尽善尽美的内核。内核社区的中心是内核邮件列表（Linux Kernel Mailing List, LKML），我们可以在 <http://vger.kernel.org/vger-lists.html#linux-kernel> 上面看到订阅这个邮件列表的细节。

内核邮件列表的流量很大，每天都有几百条消息，这里是老牛们的战场，小牛们的天堂，任何一个内核开发者都可以从中受益非浅。

除了 LKML，大多数子系统也有自己独立的邮件列表来协调各自的开发工作，比如 USB 子系统的邮件列表可以在 <http://www.linux-usb.org/mailling.html> 上面订阅。

其他网络资源

除了内核邮件列表，还有很多其他的论坛或网站值得我们经常关注。我们要知道，网络上不仅有兽兽和凤姐，也不仅有犀利哥和韩局长。

<http://www.kernel.org/>

可以通过这个网站上下载内核的源代码和补丁、跟踪内核 bug 等。

<http://kerneltrap.org>

Linux 和 BSD 内核的技术新闻。如果没时间跟踪 LKML，那么经常浏览 kerneltrap 是个好主意。

<http://lwn.net/>

Linux weekly news，创建于 1997 年底的一个 Linux 新闻站点。

<http://zh-kernel.org/mailman/listinfo/linux-kernel>

这是内核开发的中文邮件列表，里面活跃着很多内核开发领域的华人，比如 Herbert Xu、Mingming Cao、Bryan Wu 等。

<http://linux.chinaunix.net/>

全球最大的 Linux/Unix 中文技术社区。

Apache 服务器文件的详细配置

ChinaUnix 网友: Sandy

目前几乎所有的 linux 所有的发行版都捆绑了 Apache,配置 Apache 服务,需要用到 httpd 包.可以使用下面命令,查看是否安装此包。

```
[root@station195 ~]# rpm -q httpd
package httpd is not installed
```

本地配置有 yum 源,直接使用此命令安装...

```
[root@station195 ~]# yum install httpd
主配置文件:/etc/httpd/conf/httpd.conf
网页目录:/var/www/html
服务脚本:/etc/rc.d/init.d/httpd
执行程序:/usr/sbin/httpd
访问日志:/var/log/httpd/access_log
错误日志:/var/log/httpd/error_log
```

http.conf 配置文件主要由全局环境,主服务器配置和虚拟主机 3 个部分组成,配置语句可以放在文件中的任何地方,但为了增强文件的可读性,最好将配置语句放在相应的位置.所有选项指令均不区分大小写..Apache 默认会 在本机所有可用 IP 地址上的 TCP 80 端口监听客户端的请求.可以使用多个 Listen 语句,以便在多个地址和端口上监听请求.在做基于端口的虚拟主机的时候会用到...

serverRoot "/etc/httpd" apache 服务的根...

Errorlog 路径 -->定义错误日志文件..

customlog 路径 -->定义访问日志文件..

serverAdmin EMAIL -->定义管理员的邮箱..

servername IP -->定义主机名称,如果服务器有域名,则填写其域名,如果没有则写其 IP...

Timeout 连接超时..

keepalive 一次连接,多次请求,此选项默认是 off 可以置为 on.

maxkeepaliveRequests 100 当下载资源超过 100 个时断开重新连一次.

keepalivetimeout 两次请求之间间隔不能超过 15 秒.

DocumentRoot -->定义网页存放的位置.

Allowoverride None 基于身份的验证...

options Indexes Multiviews -->

Indexes 索引..允许目录浏览,当访问不到页面时,会看到目录中的文件和子目录列表.

MultiViews 允许内容协商的多重视图..

All 包含了除 MultiViews 之外的所有特性,如果没有 options 语句,默认为 All

ExecCGI 允许在该目录下执行 CGI 脚本..

FollowSymLinks 可以在该目录中使用符号链接.

Includes 允许服务器端包含的功能.

IncludesNoExec 允许服务器端包含功能,但禁止使用 CGI 脚本.

DirectoryIndex 这儿定义默认主页..

loglevel 日志级别.

细心的用户可能会发现虽然在主目录设置了 Indexes 权限,且主目录中并不存在默认文档,但访问时并不会出现目录列表,而只出现 Apache 的测试面.解决这个问题的方法是将位于/etc/httpd/conf.d 目录下的 welcome.conf 文件删除重启 Apache 即可.

AddDefaultCharset 编码 -->定义服务器返回给客户机的默认字符集..默认是西欧(UTF-8),让其支持中文使用 GB2312

最关键的一点 Order allow,deny,没有写的规则默认是 deny,如果 allow 中定义的和 deny 中定义

的规则冲突,以默认 deny 为佳...

试验一: 我们去做个试验...建一个用户 sandy,密码为 redhat.允许发布个人站点.且试验过程中开启 selinux.

```
[root@station195 conf]# vim httpd.conf
# UserDir disable
    UserDir public_html
<Directory /home/*/public_html>
    allowoverride none
    options Indexes
    order allow,deny
    allow from all
</Directory>
[root@station195 ~]# cd /home/sandy;mkdir public_html;cd public_html
[root@station195 ~]# vim index.html
welcome sandy...
[root@station195 home]# chmod o+x sandy/
[root@station195 sandy]# ls -Zld public_html/ -->查看 public_html 的标签.
drwxr-xr-x root root root:object_r:user_home_t public_html/
[root@station195 sandy]# chcon -R -t httpd_sys_content_t public_html/ -->修改此目录标
签.
[root@station195 ~]# service httpd restart
[root@station195 ~]# chkconfig httpd on
```

下面我们进行测试,在浏览器中输入 <http://192.168.0.195/~sandy> 就可以访问到网页了.....
每次访问时都要加个"~"符号,很烦人,我们可以去做个别名...

```
[root@station195 conf]# vim httpd.conf
Alias /sandy/ "/home/sandy/public_html/"
[root@station195 ~]# service httpd restart -->每次修改过配置文件都要重新启动服务器...
```

下面我们再次测试...在浏览器中输入 <http://192.168.0.195/sandy/> 就可以访问到站点了...

试验二: 基于密码验证....

```
[root@station195 conf]# vim httpd.conf
<Directory /home/*/public_html>
    allowoverride Authconfig
    options Indexes
    order allow,deny
    allow from all
</Directory>
[root@station195 ~]# cd /home/sandy/public_html;vim .htaccess
```

```
AuthName "my secret garden"
```

```
AuthType basic
```

```
AuthUserFile /etc/httpd/conf/.htpasswd
```

```
Require user natasha -->允许 natasha 用户访问.
```

```
[root@station195 ~]# htpasswd -cm /etc/httpd/conf/.htpasswd natasha -->创建 natasha  
用户给其密码为 redhat.记住在第一次创建用户的时候使用-cm 选项,第二次创建用户时使用-m 就可  
以了,因为第一次 创建时已经把.htpasswd 文件创建好了...
```

```
[root@station195 ~]# service httpd restart -->重新启动服务,这一步一定不要忘了.
```

下面去进行测试.在浏览器中输入 <http://192.168.0.195/sandy/> 会出现一个输入用户名和密码的方框,我们输入 natasha,密码 redhat,然后就可以访问到 sandy 的个人站点了....so easy!

试验三: 虚拟主机.

虚拟主机目前有三种:一种是基于端口的,一种基于 IP 的,还有一种基于主机头的...

最常用的就是基于主机头的..基于 IP 的,会耗费大量资金,基于端口的访问起了非常的不方便...下面我们去建一个站点 www.example.com

```
[root@station195 ~]# cd /etc/httpd/conf;vim httpd.conf
```

```
#DocumentRoot "/var/www/html" -->这一步很重要,首先注释掉主机头...
```

```
NameVirtualHost *:80 -->打开监听的 80 端口.
```

```
<virtualhost *:80>
```

```
serveradmin www.example.com
```

```
DocumentRoot /var/www/www
```

```
Errorlog /var/log/httpd/www/err.log
```

```
customlog /var/log/httpd/www/custom.log combined
```

```
</virtualhost>
```

```
[root@station195 httpd]# cd /var/log/httpd;mkdir www
```

```
[root@station195 www]# cd /var/www/;mkdir www;cd www
```

```
[root@station195 www]# vim index.html
```

```
this is main web!
```

```
[root@station195 ~]# vim /etc/resolv.conf
```

```
192.168.0.195 www.example.com
```

由于我本地妹做 DNS 服务器.为达到试验效果.测试机器上在 hosts 文件中也得写入这么一句才行...

```
[root@station195 ~]# service httpd restart
```

下面进行测试....在客户机浏览器 <http://www.example.com> 测试结果是成功的...大家可以模仿着做试试...

基于 https 的配置...先看 CA 证书的配置...

```
[root@station195 ~]# cd /etc/pki/CA
```

```
[root@station195 CA]# openssl genrsa 1024 > private/cakey.pem
```

```
[root@station195 CA]# vim /etc/pki/tls/openssl.cnf 将[CA_default]后 dir =../CA 改为 dir  
= /etc/pki/CA
```

```
[root@station195 CA]# mkdir ./newcerts
```



```
[root@station195 CA]# touch ./{serial,index.txt}
[root@station195 CA]# echo "00" > ./serial
[root@station195 CA]# chmod 600 private/cakey.pem
[root@station195 CA]# chmod 600 cacert.pem
```

下面进行 ssh 的配置.可以去了解[证书](#)的配置..

```
[root@station195 ~]# yum -y install mod_ssl
[root@station195 ~]# cd /etc/httpd/;mkdir ssl;cd ssl
[root@station195 ssl]# mkdir private
[root@station195 ssl]# openssl genrsa 1024 > private/httpd.key
[root@station195 ssl]# openssl req -new -key private/httpd.key -out httpd.csr
[root@station195 ssl]# openssl ca -in httpd.csr -out httpd.crt -days 3600
[root@station195 ssl]# vim ../conf.d/ssl.conf
<virtualhost 192.168.0.195:443>
servername www.example.com
documentroot /var/www/www
SSLEngine on
SSLCertificateFile /etc/httpd/ssl/httpd.crt
SSLCertificateKeyFile /etc/httpd/ssl/private/httpd.key
SSLCipherSuite ALL:!ADH:!EXPORT:!SSLv2:RC4+RSA:+HIGH:+MEDIUM:+LOW
</virtualhost>

[root@station195 ssl]# chmod 600 httpd.crt
[root@station195 ssl]# chmod 600 private/httpd.key
[root@station195 ssl]# service httpd restart
```

下面进行测试在客户端浏览器中输入 <https://www.example.com> 测试访问时成功的...这些我都严格测试过的.有一点大家注意,ssl.conf 文件中在虚拟主机部分可以定义日志文件,但日志文件要和 httpd.conf 文件中的路径和名字一样,否则报错.我开始遇到这个问题,最后解决了.像 SSLEngine SSLCertificateFile SSLCertificateKeyFile 这些都得必须定义.大家可以模仿着试试。

centos 5.4+Layer7 封 qq、msn 和 p2p 下载

ChinaUnix 网友: eric_mg

一、安装前准备

1.所需要软件包

linux-2.6.28.tar.gz <http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.28.tar.gz>
iptables-1.4.3.2.tar.bz2
[url]<http://netfilter.org/projects/iptables/files/iptables-1.4.3.2.tar.bz2>[/url]
netfilter-layer7-v2.22.tar.gz <http://sourceforge.net/projects/l7-filter/files/l7-filter%20kernel%20version/2.22/netfilter-layer7-v2.22.tar.gz/download>
l7-protocols-2009-05-28.tar.gz <http://sourceforge.net/projects/l7-filter/files/Protocol>

[%20definitions/2009-05-28/I7-protocols-2009-05-28.tar.gz/download](#)

2.把以上内核和软件包解压到/usr/src 下(这个是我个人的习惯)

```
tar -zxvf linux-2.6.28.tar.gz -C /usr/src
tar -zxvf netfilter-layer7-v2.22.tar.gz -C /usr/src
tar -zxvf I7-protocols-2009-05-28.tar.gz -C /usr/src
tar -jxvf iptables-1.4.3.2.tar.bz2 -C /usr/src
```

3.卸载系统自带的 iptables

```
rpm -qa | grep iptables  列出已安装的 iptables 包
rpm -e --nodeps ~~
```

二、安装 layer7

1.给新内核安装 layer7 补丁

```
cd /usr/src/linux-2.6.28/

patch -p1 < /usr/src/netfilter-layer7-v2.22/kernel-2.6.25-2.6.28-layer7-2.22.patch
```

2.编译新内核

2.1 修改内核配置项

```
make oldconfig 全部保持默认
make menuconfig
General setup --->
  Prompt for development and/or incomplete code/drivers 必选
Networking --->
  Networking options --->
    Network packet filtering framework (Netfilter) --->
      Core Netfilter Configuration ---> 该项下的所有项目建议都选上
        <M> Netfilter connection tracking support 这个项目必需选上，下面才会出现
layer7 的选项
          <M> "layer7" match support 必选
            Layer 7 debugging output 必选
IP: Netfilter Configuration ---> 该项下的所有项目必需都选上
```

2.2 编译并安装新内核

```
make
make modules_install
make install
```

2.3 设置新内核为默认启动的内核，如果是远程连接服务器，必须要修改这项，否则重启后默认加载旧的 kernel

```
vim /boot/grub/grub.conf
```

default=1 把 1 改为 0

timeout=15 设置等候时间

```
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.28)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.28 ro root=LABEL=/ rhgb quiet
    initrd /boot/initrd-2.6.28.img
title CentOS (2.6.18-164.el5)
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18-164.el5 ro root=LABEL=/ rhgb quiet
    initrd /boot/initrd-2.6.18-164.el5.img
```

保存退出
reboot

3.编译安装 iptables 并支持 layer7

```
cp /usr/src/netfilter-layer7-v2.22/iptables-1.4.3forward-for-kernel-2.6.20forward/*.*
extensions/
./configure --with-ksource=/usr/src/linux-2.6.28
make
make install
```

4.安装 I7 协议

```
cd /usr/src/iptables-1.4.3.2/I7-protocols-2009-05-28
make install
```

5.测试

```
iptables -V
回显：
iptables v1.4.3.2
```

```
iptables -m layer7 --help
```

回显：

```
Usage: iptables -[AD] chain rule-specification [options]
iptables -I chain [rulenum] rule-specification [options]
iptables -R chain rulenum rule-specification [options]
iptables -D chain rulenum [options]
iptables -[LS] [chain [rulenum]] [options]
iptables -[FZ] [chain] [options]
iptables -[NX] chain
iptables -E old-chain-name new-chain-name
iptables -P chain target [options]
iptables -h (print this help information)
```

Commands:

Either long or short options are allowed.

--append -A chain Append to chain
--delete -D chain Delete matching rule from chain
--delete -D chain rulenum
 Delete rule rulenum (1 = first) from chain
--insert -I chain [rulenum]
 Insert in chain as rulenum (default 1=first)
--replace -R chain rulenum
 Replace rule rulenum (1 = first) in chain
--list -L [chain [rulenum]]
 List the rules in a chain or all chains
--list-rules -S [chain [rulenum]]
 Print the rules in a chain or all chains
--flush -F [chain] Delete all rules in chain or all chains
--zero -Z [chain] Zero counters in chain or all chains
--new -N chain Create a new user-defined chain
--delete-chain
 -X [chain] Delete a user-defined chain
--policy -P chain target
 Change policy on chain to target
--rename-chain
 -E old-chain new-chain
 Change chain name, (moving any references)

Options:

[!] --proto -p proto protocol: by number or name, eg. 'tcp'
[!] --source -s address[/mask]
 source specification
[!] --destination -d address[/mask]
 destination specification
[!] --in-interface -i input name[+]
 network interface name ([+] for wildcard)
--jump -j target
 target for rule (may load target extension)
--goto -g chain
 jump to chain with no return
--match -m match
 extended match (may load extension)
--numeric -n numeric output of addresses and ports
[!] --out-interface -o output name[+]
 network interface name ([+] for wildcard)
--table -t table table to manipulate (default: 'filter')
--verbose -v verbose mode
--line-numbers print line numbers when listing

```
--exact -x          expand numbers (display exact values)
[!] --fragment -f      match second or further fragments only
--modprobe=<command>   try to insert modules using this command
--set-counters PKTS BYTES set the counter during insert/append
[!] --version -V        print package version.
```

layer7 match options:

```
--l7dir <directory> : Look for patterns here instead of /etc/l7-protocols/
                      (--l7dir must be specified before --l7proto if used)
[!] --l7proto <name>: Match named protocol using /etc/l7-protocols/.../name.pat
```

至此安装过程完全完成，如果有问题，请通过 ChinaUnix 论坛站内短信联系 eric_mg。

Linux 上 Postgres Plus 数据库备份和恢复

IT168 网友：宇文

本文将为读者介绍如何使用 pg_dump 和 pg_restore 来保护 Postgres Plus 数据库。我们会向您介绍如何构建数据库和应用程序，怎样周期性创建数据库备份的方法，并在需要的时候修复它们。

一、引言

本文将为读者详解在 linux 环境中快速玩转 Postgres Plus Standard Server 或者 Postgres Plus Advanced Server 数据库产品的必要知识，在此之前，我们假定您已经下载并且安装好了 Postgres Plus Standard Server 或者 Postgres Plus Advanced Server。这将有助于您加快对于 Postgres Plus Standard Server 或者 Postgres Plus Advanced Server 的技术评估。

通过本文，您将能够掌握下列技能：

- 熟悉不同的备份格式
- 选择不同的备份和恢复选项
- 使用纯文本进行备份和恢复
- 为备份和恢复创建一个定制的归档

二、功能描述

在备份和恢复 Postgres Plus 数据库的时候，我们有多种不同的方法和选项可用，本文将向您展示如何利用两个 Postgres Plus 实用程序，即 pg_dump 和 pg_restore 来完成数据库的备份和恢复。这两个程序可以在命令行中使用，所以只要您喜欢的话，完全可以将它们集成到您自己的脚本之中。

注意：有关使用 pg_dump 创建一个备份文件更详细的说明，请参阅 Postgres Plus web 文档页面中的“PostgreSQL Client Applications”一章。有关使用 pg_restore 恢复一个备份文件更详细的说明，请参阅 Postgres Plus web 文档页面中的“PostgreSQL Client Applications”一章。有关 Postgres Plus 数据库所有备份和恢复策略的说明，请参阅 Postgres Plus web 文档页面中的“backup and restore”一章。备份文件的格式

利用 pg_dump，我们可以创建三种不同的备份文件格式，如下所示：

- 纯文本格式。纯文本格式的脚本文件，可以包含 SQL 语句和命令，然后用命令行终端程序 psql 运行该文件，来重新创建数据库对象并装入表中数据。我们可以使用 psql 程序从纯文本格式的备份文件中进行恢复操作。

- 定制的归档格式。它是一个二进制文件，能够用来从备份文件恢复选定的数据库对象或者全部对象。我们可以使用 pg_restore 程序从定制的归档格式的备份文件中完成恢复。

- Tar 归档格式。Tar 归档文件可以从备份文件中恢复选取的数据库对象或全部数据库对象。我们可以使用 pg_restore 程序从 Tar 归档格式的备份文件中进行恢复。

如果您需要的话，可以在用 psql 程序恢复纯文本格式的备份文件的数据库对象之前，使用文字编辑器编辑该纯文本文件。纯文本格式适用于小型的数据库。

定制的归档备份文件无法进行编辑，不过我们可以使用 pg_restore 程序从备份文件中选择需要恢复的数据库对象。定制的归档格式通常用于中到大型的数据库并且希望从备份文件中有选择性地恢复数据库对象的情形。

Tar 归档备份文件可以使用诸如 tar 之类的标准 Linux 工具进行操作。就像定制的归档格式一样，我们也可以使用 pg_restore 程序从备份文件中选择需要恢复的数据库对象。

您可以使用 pg_dump 程序来压缩纯文本或者定制的归档格式的备份文件，这样能够缩小备份文件的尺寸。当生成纯文本格式的备份文件时，默认的配置为不进行压缩。当生成定制的归档格式的备份文件时，默认配置为进行中等水平的压缩处理。pg_dump 程序无法对 Tar 归档格式的备份文件进行压缩处理。

三、与备份和恢复有关的选项

Pg_dump 和 pg_restore 程序为我们提供了许多选项，通过它们可以控制将哪些数据库对象保存到备份文件，从备份文件中恢复哪些数据库对象，以及如何进行恢复。下面列出了一些选项：

- 仅转储或者恢复模式 (表、视图和序列定义、约束、触发器与函数)，不转储或恢复表中数据。(如果您当前使用的是 Postgres Plus Advanced Server 的话，SPL 函数、过程、触发器和[软件包](#)也是能够进行备份和恢复的。)

- 仅转储或恢复表中数据，而不包括模式。

- 转储或恢复属于选中的模式的数据库对象。

- 创建备份时，排除选中的方案。

- 转储或恢复选中的表。

- 创建备份时，排除选中的表。

- 允许恢复操作使用要备份的数据库的名称来新建数据库，然后从这个新创的数据库中恢复数据库对象。

- 将数据库对象恢复到所有的现有数据库。

- 令被恢复的数据库对象的属主与制作备份时这些对象的属主同名。

- 将进行恢复操作的用户的角色指定为所有被恢复的数据库对象的属主。

注意：有关创建一个备份文件更详细的说明，请参阅 Postgres Plus web 文档页面中的“PostgreSQL Client Applications”一章中关于 pg_dump 的内容。有关恢复定制的归档格式的备份文件的详细过程，请参阅 Postgres Plus web 文档页面中的“PostgreSQL Client Applications”一章中关于 pg_restore 的内容。

假设我们想要备份一个数据库的全部内容，并在随后某个时间从备份文件中重建整个数据库，下面我们就介绍用纯文本格式的备份文件和定制的归档格式的备份文件来完成上述过程。

四、指导步骤

假定您的数据库环境如下所示：

- 我们的数据库要恢复到其中的数据库集群包含的角色名称(用户名和组名)就是制作备份时数据库对象的属主。换句话说，我们要制作备份(当制作备份时没有删除任何数据库对象的属主的角色)的数据库集群与恢复的目的地是同一个数据库集群，或者我们要恢复到一个新的数据库集群，并且在其中添加的一组角色名称与制作备份的数据库集群中的角色名称完全相同。

- 用来恢复我们的数据库的数据库集群中没有与制作备份的数据库同名的数据库。(如果制作备份文件的数据库集群正好是用来恢复数据库的数据库集群，则数据库制作备份之后，需要将其删除或者改名。)

注意：一个数据库集群就是由同一个 Postgres Plus 实例运行的一组数据库。一个数据库集群通过它的 ip 地址和端口号唯一标识。

五、使用纯文本格式的备份文件备份数据库

下面我们详细介绍如何使用 pg_dump 程序创建数据库的纯文本格式的备份文件。

第一步：登录到 Postgres Plus 数据库[服务器](#)所在的计算机。在该计算机上任何有效的帐户都可以。

第二步：使用 cd 命令将当前工作目录切换至 Postgres Plus 所在的 bin 目录。

```
cd /opt/PostgresPlus/8.4SS/bin
```

注意：如果您使用的是 Postgres Plus Advanced Server 的话，请将当前工作目录通过 cd 命令切换至 dbserver/bin。

第三步：使用具有超级用户权限(-U 选项)的角色运行 pg_dump 程序，使用 -C 来包含备份文件中的 CREATE DATABASE 语句，给备份文件指定的名称(-f 选项)，以及用来制作备份文件的数据库的名称(命令行的最后一个参数)。

```
./pg_dump -U postgres -C -f /home/user/sample_backup sample
```

上面的命令就会为数据库 sample 建立一个备份文件，名为 sample_backup。下面是 sample_backup 文件的一部分内容：

```
--
-- PostgreSQL database dump
--
SET statement_timeout = 0;
```

```
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;
--
-- Name: sample; Type: DATABASE; Schema: -; Owner: postgres
--
CREATE DATABASE sample WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE
= 'en_US.UTF-8' LC_CTYPE = 'en_US.UTF-8';
ALTER DATABASE sample OWNER TO postgres;
\connect sample
SET statement_timeout = 0;

SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;
--
-- Name: plpgsql; Type: PROCEDURAL LANGUAGE; Schema: -; Owner: postgres
--
CREATE PROCEDURAL LANGUAGE plpgsql;
ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
SET search_path = public, pg_catalog;
```

六、从纯文本格式的备份文件恢复数据库

下面我们详细介绍如何使用 psql 程序从一个纯文本格式的备份文件恢复数据库。

注意：如果您使用的是 Postgres Plus Advanced Server 的话，那么可以使用 edb-psql 程序。

上面，我们为数据库 sample 建立了一个纯文本格式的备份文件，即 sample_backup，之后我们要用它来恢复该示例数据库。

第一步：登录到 Postgres Plus 数据库[服务器](#)所在的计算机。在该计算机上任何有效的帐户都可以。

第二步：如果想把数据库恢复到建立备份文件的数据库集群之外的数据库集群的话，或者从数据库集群中删除了角色的话，那么要确保制作备份文件时数据库对象属主的所有的角色名称都存在于用来恢复备份文件的数据库集群中。

注意：如果制作备份文件时不知道数据库对象属于哪些角色，您可以使用文本编辑程序搜索 ALTER object OWNER TO role 语句，下面是些例子：

```
ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
```

```
ALTER TYPE public.emp_query_type OWNER TO postgres;
ALTER FUNCTION public.emp_comp(p_sal numeric, p_comm numeric) OWNER TO
postgres;
```

您可以通过 psql 程序连接到数据库集群，然后运行 \dg 命令列出当前存在于数据库集群中的角色，如下所示：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.
postgres=# \dg
List of roles
Role name | Attributes | Member of
-----+-----+-----
postgres | Superuser | {}
: Create role
: Create DB
```

如果您需要创建角色的话，可以使用 CREATE ROLE 语句。

注意：如果用来恢复数据库对象的数据库集群中没有该数据库对象原属主的角色，那么当 ALTER 语句无法指定该属主时，就会显示一个出错信息。当执行恢复操作时，该数据库对象最终将分配给 psql 程序的 -U 选项指定的角色。

第三步：保证在用于恢复数据库的数据库集群中没有与制作备份的数据库同名的数据库。

注意：如果不知道制作备份文件的数据库的名称，您可以使用文本编辑程序搜索 CREATE DATABASE 语句，下面举例说明：

```
CREATE DATABASE sample WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE
= 'en_US.UTF-8' LC_CTYPE = 'en_US.UTF-8';
```

您可以通过 psql 程序连接到数据库集群，然后运行 \l 命令列出当前存在于数据库集群中的数据库，如下所示：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.
postgres=# \l
List of databases
Name | Owner | Encoding | Collation | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
```

```
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
: postgres=CTc/postgres
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
: postgres=CTc/postgres
(3 rows)
```

如果数据库集群中的一个数据库与用于建立备份文件的数据库同名，那么您现在就可以使用 ALTER DATABASE 语句的 RENAME TO 选项给现有的数据库改名，或者使用 DROP DATABASE 语句删除它。

注意：在恢复操作期间，如果该数据库集群含有一个与制作备份文件的数据库同名的数据库的话，那么当 CREATE DATABASE 语句无法用同一名称新建数据库时，就会显示一个出错信息。这时，psql 程序将从现有的数据库中的备份文件来重新创建各数据库对象。这很可能导致生成的数据库中包含不希望有的数据库对象，并且数据库设置也不正确。

第四步：使用 cd 命令将当前工作目录切换至 Postgres Plus 所在的 bin 目录。

```
cd /opt/PostgresPlus/8.4SS/bin
```

注意：如果您使用的是 Postgres Plus Advanced Server 的话，请将当前工作目录通过 cd 命令切换至 dbserver/bin。

第五步：运行 psql 程序，并给出要连接(使用 -d 选项)的数据库的名称、具有超级用户特权的角色(使用 -U 选项)和备份文件的目录路径(使用 -f 选项)。

注意：用 -d 选项规定的数据库对恢复操作没有影响。psql 程序需要先通过数据库连接建立一个会话，然后才能处理备份文件中的 SQL 语句和 psql 命令。

```
./psql -d postgres -U postgres -f /home/user/sample_backup
```

这样就会从备份文件 sample_backup 中重新创建数据库 sample。下面给出 psql 程序处理备份文件中的 SQL 语句和 psql 命令时所显示的部分消息：

```
$ ./psql -d postgres -U postgres -f /home/user/sample_backup
Password for user postgres:
SET
SET
SET
SET
SET
SET
CREATE DATABASE
ALTER DATABASE
psql (8.4.1)
You are now connected to database "sample".
```

七、创建定制的归档格式的数据库备份文件

下面我们详细介绍如何使用 pg_dump 程序创建一个数据库的定制的归档格式的备份文件。

第一步：登录到 Postgres Plus 数据库服务器所在的计算机。在该计算机上任何有效的帐户都可以。

第二步：使用 cd 命令将当前工作目录切换至 Postgres Plus 所在的 bin 目录。

```
cd /opt/PostgresPlus/8.4SS/bin
```

注意：如果您使用的是 Postgres Plus Advanced Server 的话，请将当前工作目录通过 cd 命令切换至 dbserver/bin。

第三步：运行 pg_dump 程序，使用 -U 选项给出具有超级用户权限的角色，使用 -Fc 选项规定定制的归档格式，利用 -f 选项给备份文件指定名称，并给出用来制作备份文件的数据库的名称(命令行的最后一个参数)。

```
./pg_dump -U postgres -Fc -f /home/user/sample_backup sample
```

上面的命令就会为数据库 sample 建立一个备份文件，名为 sample_backup。虽然定制的归档格式的备份文件无法直查看，但是却可以通过 pg_restore 程序的 -l 选项来生成备份文件的目录，操作步骤如下：

```
$ ./pg_restore -l /home/user/sample_backup
;
; Archive created at Thu Nov 12 15:12:12 2009
; dbname: sample
; TOC Entries: 44
; Compression: -1
; Dump Version: 1.11-0
; Format: CUSTOM
; Integer: 4 bytes
; Offset: 8 bytes
; Dumped from database version: 8.4.1
; Dumped by pg_dump version: 8.4.1
;;
; Selected TOC Entries:
;
6; 2615 2200 SCHEMA - public postgres
1818; 0 0 COMMENT - SCHEMA public postgres
1819; 0 0 ACL - public postgres
323; 2612 17798 PROCEDURAL LANGUAGE - plpgsql postgres
312; 1247 17801 TYPE public emp_query_type postgres
19; 1255 17802 FUNCTION public emp_comp(numeric, numeric) postgres
20; 1255 17803 FUNCTION public emp_query(numeric, numeric, character varying)
postgres
21; 1255 17804 FUNCTION public emp_query_caller() postgres
22; 1255 17805 FUNCTION public emp_sal_trig() postgres
```

八、从定制的归档格式的备份文件中恢复数据库

下面我们详细介绍如何使用 pg_restore 程序从一个定制的归档格式的备份文件中恢复数据库。

之前，我们从数据库 sample 建立了一个定制的归档格式的备份文件，即 sample_backup，接下来我们要用它来恢复数据库 sample。

第一步：登录到 Postgres Plus 数据库服务器所在的计算机。在该计算机上任何有效的帐户都可以。

第二步：如果想把数据库恢复到建立备份文件的数据库集群之外的数据库集群的话，或者从数据库集群中删除了角色的话，那么要确保制作备份文件时数据库对象属主的所有的角色名称都存在于用来恢复备份文件的数据库集群中。

注意：制作备份文件时如果不知道数据库对象属于哪些角色，您就可以从定制的归档格式的备份文件中生成一个 SQL 文本版本的备份。为此，运行 pg_restore 程序，以备份文件作为其唯一的参数。然后您可以在文本中搜索 ALTER object OWNER TO role 语句。具体如下所示：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./pg_restore /home/user/sample_backup | grep 'OWNER TO'
ALTER PROCEDURAL LANGUAGE plpgsql OWNER TO postgres;
ALTER TYPE public.emp_query_type OWNER TO postgres;
ALTER FUNCTION public.emp_comp(p_sal numeric, p_comm numeric) OWNER TO
postgres;
```

您可以通过 psql 程序连接到数据库集群，然后运行 \dg 命令列出当前存在于数据库集群中的角色，如下所示：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.
postgres=# \dg
List of roles
Role name | Attributes | Member of
-----+-----+-----
postgres | Superuser | {}
: Create role
: Create DB
```

如果您需要创建角色的话，可以使用 CREATE ROLE 语句。注意：如果用来恢复数据库对象的数据库集群中没有该数据库对象原属主的角色，那么当 ALTER 语句无法指定该属主时，就会显示一个出错信息。当执行恢复操作时，该数据库对象最终将分配给 pg_restore 程序的-U 选项指定的角色。

第三步：保证用于恢复数据库的数据库集群中，现有数据库不会与制作备份的数据库同名。注

意：如果不知道制作备份文件的数据库的名称，您可以使用 pg_restore 程序，加上 -l 选项，以备份文件作为唯一的参数来列出定制的归档格式的备份文件的目录。数据库名称位于目录顶部的 dbname 字段。

下面是一个例子：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./pg_restore -l /home/user/sample_backup
;
; Archive created at Wed Nov 18 12:34:10 2009
; dbname: sample
```

您可以通过 psql 程序连接到数据库集群，然后运行 \l 命令列出当前存在于数据库集群中的数据库，如下所示：

```
$ cd /opt/PostgresPlus/8.4SS/bin
$ ./psql -d postgres -U postgres
Password for user postgres:
psql (8.4.1)
Type "help" for help.
postgres=# \l
List of databases
Name | Owner | Encoding | Collation | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
postgres | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | 
template0 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
: postgres=Ctc/postgres
template1 | postgres | UTF8 | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
: postgres=Ctc/postgres
(3 rows)
```

如果数据库集群中的一个数据库与用于建立备份文件的数据库同名的话，您可以使用 ALTER DATABASE 语句的 RENAME TO 选项给现有的数据库改名，或者使用 DROP DATABASE 语句将其删除。

注意：在恢复操作期间，如果该数据库集群含有一个与制作备份文件的数据库同名的数据库的话，那么当 CREATE DATABASE 语句无法用同样的名字新建数据库时，就会显示出错信息。此后，pg_restore 程序会从现有的数据库内的从备份文件中重新创建数据库对象。这很可能导致数据库内包含不希望的数据库对象以及错误的数据库设置。

第四步：使用 cd 命令将当前工作目录切换至 Postgres Plus 所在的 bin 目录。

```
cd /opt/PostgresPlus/8.4SS/bin
```

注意：如果您使用的是 Postgres Plus Advanced Server 的话，请将当前工作目录通过 cd 命令切换至 dbserver/bin。

第五步：运行 pg_restore 程序，用 -d 选项给出要连接的数据库的名称，使用 -U 选项给出具有

超级用户权限的角色, 利用-C 选项指出本恢复操作新建的数据库与制作备份的数据库同名, 并通过命令行的最后一个参数给出备份文件的目录路径。

第六步: 运行 psql 程序, 并给出要连接(使用-d 选项)的数据库的名称、具有超级用户特权的角色(使用-U 选项)和备份文件的目录路径(使用-f 选项)。pg_restore 程序需要通过数据库连接建立一个会话, 然后才能从备份文件中创建新的数据库并恢复数据库对象。

```
./pg_restore -d postgres -U postgres -C /home/user/sample_backup
```

这样, 我们就会从名为 sample_backup 的备份文件中重新创建数据库 sample。

九、结束语

本文为读者详细介绍了在 linux 系统上备份和恢复 Postgres Plus 数据库的具体操作过程。通过备份和恢复功能, 我们能够在工作的不同阶段对数据库进行备份, 并在需要的时候进行相应的恢复操作。

alsa 音频库的移植

ChinaUnix 网友: bluedrum

alsa 是在 Linux 出现比较晚的音频处理库,主要是针对原有的 OSS 的库进行改进.性能更好,支持多个播放源等.但相应的,增加的编程的复杂性。

旧的 oss 是采用标准文件 IO 接口,设备打采用 open,声音播录是采用数据 IO 接口(read/write),设备的参数采用 ioctl 来控制.都是 系统调用因此无需额外库和配置文件来操作设备.但 alsa 也是最终采用 open/read/write/ioctl,但操作结点更多.更复杂.因此在编程 中是采用额外的库来操作的。

ALSA 的体系结构

官方主页 <http://www.alsa-project.org/>

主要跟编程相关是

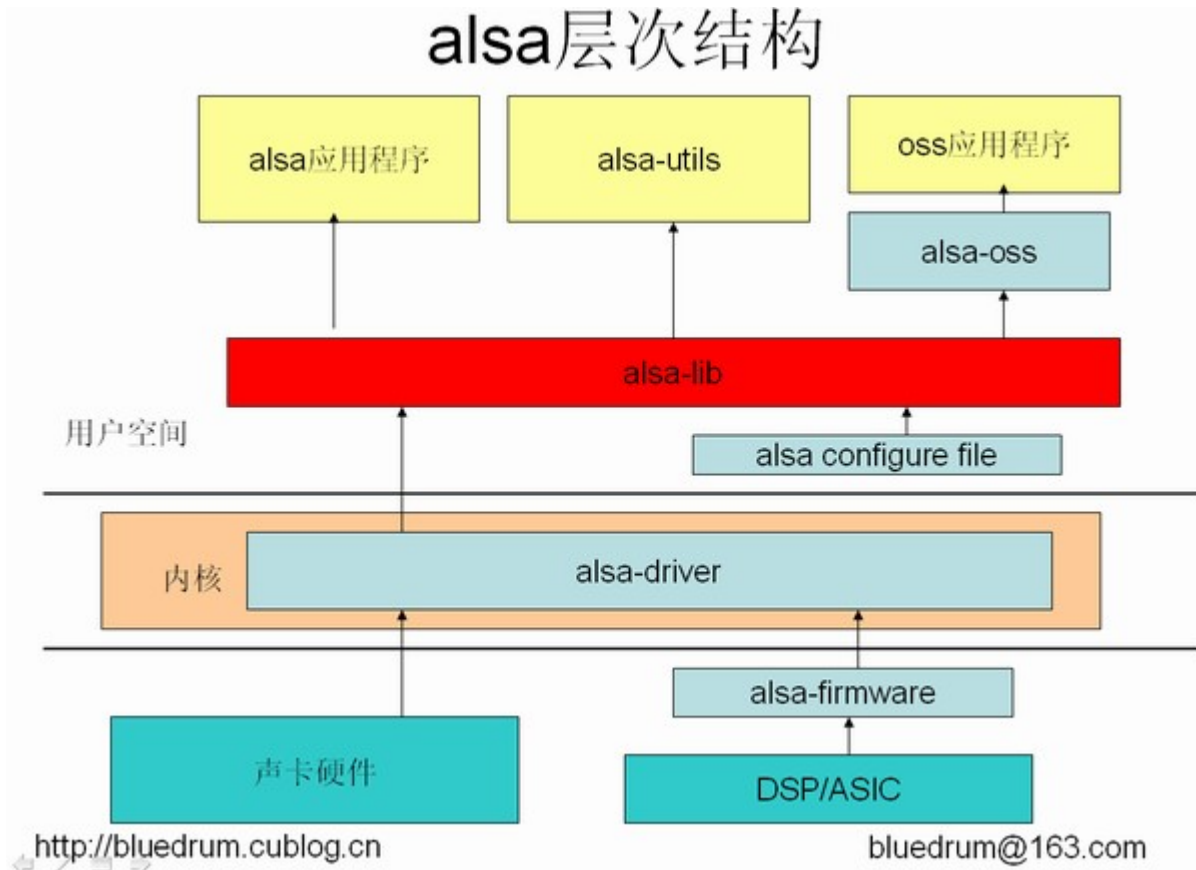
- alsa-lib. ALSA 应用库(最常用)
 - <ftp://ftp.alsa-project.org/pub/lib/alsa-lib-1.0.22.tar.bz2>
- alsa-driver 一些常见芯片的 ALSA 驱动代码,一般内核会集成.
 - <ftp://ftp.alsa-project.org/pub/driver/alsa-driver-1.0.22.1.tar.bz2>
- alsa-firmware 一些 DSP 或 ASIC 的专用的微码(运在芯片之上,启动时由 LINUX 装入到硬件中).
 - <ftp://ftp.alsa-project.org/pub/firmware/alsa-firmware-1.0.20.tar.bz2>
- alsa-utils 一般 ALSA 小的测试工具.如 aplay/arecord 播放和录音小程序.
 - <ftp://ftp.alsa-project.org/pub/utils/alsa-utils-1.0.22.tar.bz2>
- alsa-oss 用 alsa 接口模拟旧的 oss 接口.
 - <ftp://ftp.alsa-project.org/pub/oss-lib/alsa-oss-1.0.17.tar.bz2>

其中 alsa-driver,alsa-firwware 是内核开发者所接触的东西,对于已经正常运行硬件,通常意味着这

一部分已经整合到内核当中,无需修改。

而 alsa-utils 主要是测试一些小工具.

因此对于一个应用程序开发者,或者嵌入式应用开发者,接触到主要是 alsa-lib 编译出来的库 libasound.



Linux ALSA 驱动测试

以下是我在 RHEL 5 上测试 ALSA 的结果,在 ARM 或 MIPS 开发板上类似

如果安装 ALSA,在 /proc 会看到相应设备:

用 `cat /proc/asound/devices`

```
[root@hxy alsa-lib-1.0.22]# cat
/proc/asound/devices
2:      : timer
3:      : sequencer
4: [ 0- 0]: raw midi
5: [ 0- 1]: digital audio playback
6: [ 0- 0]: digital audio playback
7: [ 0- 0]: digital audio capture
8: [ 0]  : control
```


正常情况下,在你的/dev/snd 会看到如下设备结点(有例外,就是内核驱动调整了结点位置)

用 `ls -l /dev/snd`

```
[root@hxy alsa-lib-1.0.22]# ls -l /dev/snd
total 0
crw----- 1 root root 116, 8 Mar  2 02:41 controlC0
crw----- 1 root root 116, 4 Mar  2 02:41 midiC0D0
crw----- 1 root root 116, 7 Mar  2 02:41
pcmC0D0c
crw----- 1 root root 116, 6 Mar  2 02:41
pcmC0D0p
crw----- 1 root root 116, 5 Mar  2 02:41
pcmC0D1p
crw----- 1 root root 116, 3 Mar  2 02:41 seq
crw----- 1 root root 116, 2 Mar  2 02:41 timer
```

通常桌面版已经安装了 alsa-utils.可以执行一下 `aplay -h`,测试是否输出.如果可以来播入一下 wav 文件来测试整个环境是否正常工作。

以下是一些用 alsa-utils 测试样例

- 播放 wave 文件
 - `aplay /mnt/nfs/test.wav`
- 变频播放,(以是以 44 KHz 来播放音频)
 - `aplay -D rate_44k /mnt/nfs/test.wav`
- 录音,以 20 秒的间隔(-d 20),立体声(-c 2),频率是 8000Hz 来录制 Wave 格式音频
 - `arecord -d 20 -c 2 -t wav -r 8000 -f "Signed 16 bit Little Endian" /mnt/nfs/test.wav`
- 测试混音播放(先是播放 test1.wav,然后再同时播放 test2.wav)
 - `aplay -D plug:dmix_44k /mnt/nfs/test1.wav &`
 - `aplay -D plug:dmix_44k /mnt/nfs/test2.wav`
- 设置放音增益 (0 to 3)
 - `amixer set Master 1`
- 设置录音音量(0-31)
 - `amixer set Line 10`

嵌入式 LINUX 下 alsa 库移植.

在 ARM-LINUX/MIPS LINUX 下移植 ALSA 库,首先保存设备驱动结点正确无误。

然后主要移植 alsa-lib

alsa-lib 的移植.

解压 tar xvjf alsa-lib-1.0.22.tar.bz2

cd alsa-lib-1.0.22

生成 Makefile

```
./configure --host=arm-linux --prefix=$PWD/../../output/arm-linux --enable-static --enable-shared --disable-python --with-configdir=/usr/local/share --with-plugindir=/usr/local/lib/alsa_lib
```

在这里要注意 --with--configdir 的选项.它将影响 include/config.h 中的 ALSA_CONFIG_DIR 目录。它默认是你的--prefix 目录.这样在嵌入式交叉编译将是一个桌面机的路径,在 libasoud.so 运行.会提示,如果出来这个提示,一般都是 ALSA_CONFIG_DIR 路径错误造成的。

```
ALSA lib pcm.c:2145:(snd_pcm_open_noupdate) Unknown PCM
default
```

```
aplay: main:546: audio open error: No such file or directory
```

--with-plugindir 也是同样道理了.它是设为 ALSA_PLUGIN_DIR 宏.

编译 make

安装 make install

开发板发布注意:

在开发板上发布 alsa 库.除了 libasound.so 库以外,必须还要把 alsa.conf 发布到板上--with-configdir 所指向目录下的 alsa 目录,否则还是会报"audio open error: No such file or directory".

这个文件可以在 make install 后在你安装目录下的 share 找到 alsa 目录.把这个目录整个拷贝到开发板即可.

alsa-utils 移植

解压:tar xvjf alsa-utils-1.0.22.tar.bz2

cd alsa-utils-1.0.22

生成 Makefile

```
./configure --host=arm-linux --prefix=$PWD/../../output/arm-linux --enable-static --enable-shared --with-configdir=/usr/local/share --with-libiconv-prefix=$PWD/../../output/arm-linux CFLAGS="-I$PWD/../../output/arm-linux/include" LDFLAGS="-L$PWD/../../output/arm-linux/lib -lasound -liconv" --disable-alsamixer --disable-xmlto
```

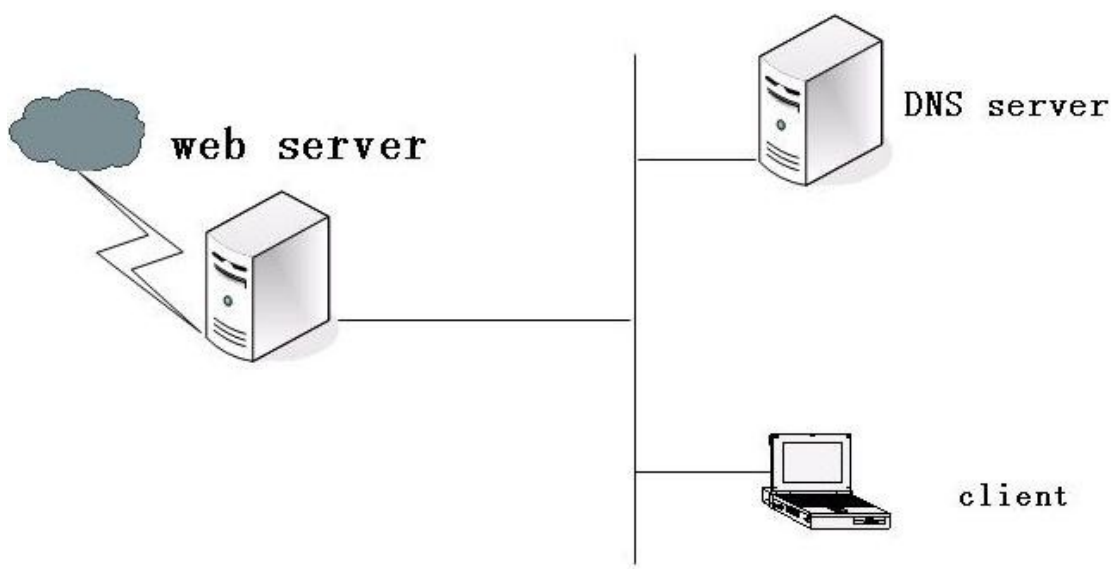
注意这里 LDFLAGS 是必须,否则会找不到 libasound.另外 alsamixer 是一个 ncurses 程序,基本上在嵌入式终端上很难移植.所以这里取消掉.--disable-xmlto 也是因为找不到库。

编译 make

安装 make install

RHEL5. 4 下 LAMP 平台搭建及基本测试

ChinaUnix 网友: crazylinux



实验目的:

完成 Linux 下面的 LAMP 环境的搭建, 实现 Discuz 论坛, 实现 awstats 分析 Web 日志, 以及 AB 实现 HTTP 性能测试, 最后通过 phpMyAdmin 管理 MySQL 数据库

实验软件包:

软件下载地址: [ChinaUnix 软件下载](#)

实验环境: RHEL5.4

Web server: 192.168.0.10/24

Dns server: 192.168.0.20/24

Clinet: 192.168.0.100/24

前提条件:

在所有 linux 主机上配置 yum 仓库, 把 DVD 光盘挂在到本地/media 下, 红色字体代表修改后结果

```
[root@localhost ~]# cat /etc/yum.repos.d/rhel-debuginfo.repo
```

```
[rhel-debuginfo]
```

```
name=Red Hat Enterprise Linux $releasever - $basearch - Debug
```

```
baseurl=file:///media/Server    //指定本地/media
enabled=1                      //开启仓库
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

关闭所有 linux 防火墙与 SELINUX

```
[root@localhost bin]# iptables -F
[root@localhost bin]# iptables -X
[root@localhost bin]# iptables -Z
[root@localhost bin]# setenforce 0
[root@localhost bin]# service iptables save
```

1、DNS 环境的搭建

1. 安装 DNS 相关软件包

```
[root@localhost ~]# yum -y install bind bind-devel bind-chroot caching-nameserver
```

2、配置 DNS 主配置文件

```
[root@localhost ~]# cd /var/named/chroot/etc/
[root@localhost etc]# cp -p named.caching-nameserver.conf named.conf
[root@localhost etc]# cat named.conf
//
// named.caching-nameserver.conf
//
// Provided by Red Hat caching-nameserver package to configure the
// ISC BIND named(8) DNS server as a caching only nameserver
// (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// DO NOT EDIT THIS FILE - use system-config-bind or an editor
// to create named.conf - edits to this file will be lost on
// caching-nameserver package upgrade.
//
options {
```

```
listen-on port 53 { any; };
listen-on-v6 port 53 { ::1; };
directory    "/var/named";
dump-file    "/var/named/data/cache_dump.db";
statistics-file "/var/named/data/named_stats.txt";
memstatistics-file "/var/named/data/named_mem_stats.txt";

// Those options should be used carefully because they disable port
// randomization
// query-source    port 53;
// query-source-v6 port 53;

allow-query    { any; };
allow-query-cache { any; };
};
logging {
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};
view localhost_resolver {
    match-clients    { any; };
    match-destinations { any; };
    recursion yes;
    include "/etc/named.rfc1912.zones";
};
[root@localhost etc]# vim named.rfc1912.zones 在文件最后添加
zone "crazylinux.com" IN {
    type master;
    file "cl.zheng";
```



```
};
```

```
zone "google.com" IN {  
    type master;  
    file "g.zheng";  
};
```

```
zone "0.168.192.in-addr.arpa" IN {  
    type master;  
    file "cl.fan";  
};
```

```
};
```

3、配置区域文件

```
[root@localhost etc]# cd /var/named/chroot/var/named/
```

```
[root@localhost named]# cp -p named.zero cl.zheng
```

```
[root@localhost named]# cp -p named.zero g.zheng
```

```
[root@localhost named]# cp -p named.zero cl.fan
```

```
[root@localhost named]# cat cl.zheng
```

```
$TTL 86400
```

```
@      IN SOA localhost.  root.localhost. (  
                42      ; serial (d. adams)  
                3H      ; refresh  
                15M     ; retry  
                1W      ; expiry  
                1D )    ; minimum
```

```
IN NS dns.crazylinux.com.
```

```
dns IN A 192.168.0.20
```

```
www IN A 192.168.0.10
```

```
[root@localhost named]# cat g.zheng
```

```
$TTL 86400
```

```
@      IN SOA localhost.  root.localhost. (  
                42      ; serial (d. adams)
```

```

        3H      ; refresh
        15M     ; retry
        1W      ; expiry
        1D )    ; minimum
    IN  NS  dns.google.com.
dns  IN  A   192.168.0.20
www  IN  A   192.168.0.10
[root@localhost named]# cat cl.fan
$TTL  86400
@      IN SOA localhost.  root.localhost. (
        42      ; serial (d. adams)
        3H      ; refresh
        15M     ; retry
        1W      ; expiry
        1D )    ; minimum
    IN  NS  dns.crazylinux.com.
20  IN  PTR  dns.crazylinux.com.
20  IN  PTR  dns.google.com.
10  IN  PTR  www.crazylinux.com.
10  IN  PTR  www.google.com.
[root@localhost named]# cat /etc/resolv.conf
nameserver 192.168.0.20
search example.com
[root@localhost named]# service named restart
[root@localhost named]# chkconfig named on
4、测试 DNS
[root@localhost named]# host www.crazylinux.com
www.crazylinux.com has address 192.168.0.10
[root@localhost named]# host www.google.com
www.google.com has address 192.168.0.10
```

2. LAMP 环境的搭建

1、安装 APACHE（配置源码包之前先安装 GCC 环境）

```
[root@localhost ~]# yum -y install gcc
[root@localhost bin]# rpm -e httpd --nodeps
[root@localhost media]# tar -jxvf httpd-2.2.13.tar.bz2 -C /usr/local/src/
[root@localhost httpd-2.2.13]# ./configure --prefix=/usr/local/apache2
--enable-so --enable-rewrite
[root@localhost httpd-2.2.13]# make && make install
```

做个小技巧，实现开机自动启动 APACHE

```
[root@localhost httpd-2.2.13]# cd /usr/local/apache2/bin/
[root@localhost bin]# vim apachectl
#!/bin/sh
#
# chkconfig: - 85 15
# description: Apache is a World Wide Web server. It is used to serve \
#              HTML files and CGI.
[root@localhost bin]# cp apachectl /etc/init.d/apache
[root@localhost bin]# chkconfig --add apache
[root@localhost bin]# service apache restart
[root@localhost bin]# chkconfig apache on
```

测试：在 cline 上 IE 中输入服务器的 ip

<http://192.168.0.10>

1) 配置个人主页

```
[root@localhost bin]# vim /usr/local/apache2/conf/httpd.conf 把 382 行前的 # 去掉
381 # User home directories
382 Include conf/extra/httpd-userdir.conf
[root@localhost bin]# useradd crazylinux
[root@localhost bin]# su - crazylinux
[crazylinux@localhost ~]$ mkdir public_html
[crazylinux@localhost ~]$ echo "Crazylinux's Home Page" >> public_html/index.html
```

```
[crazylinux@localhost ~]$ chmod o+x /home/crazylinux/  
[crazylinux@localhost ~]$ exit  
[root@localhost bin]# service apache restart
```

去掉~

```
[root@localhost bin]# vim /usr/local/apache2/conf/extra/httpd-userdir.conf  
在第 10 行下添加  
10 UserDir public_html  
11 Alias /crazylinux "/home/crazylinux/public_html"  
[root@localhost bin]# service apache restart
```

2) 构建基于域名的虚拟主机

前提条件

```
[root@localhost bin]# cat /etc/resolv.conf  
nameserver 192.168.0.20  
search example.com  
[root@localhost bin]# host www.crazylinux.com  
www.crazylinux.com has address 192.168.0.10  
[root@localhost bin]# host www.google.com  
www.google.com has address 192.168.0.10
```

开始配置基于域名的虚拟主机

```
[root@localhost bin]# vim /usr/local/apache2/conf/httpd.conf 把 388 行前面的#号去掉  
387 # Virtual hosts
```

```
388 Include conf/extra/httpd-vhosts.conf
```

```
[root@localhost bin]# vim /usr/local/apache2/conf/extra/httpd-vhosts.conf 修改下面这样
```

```
NameVirtualHost 192.168.0.10
```

```
<VirtualHost 192.168.0.10>
```

```
    DocumentRoot "/usr/local/apache2/htdocs/crazylinux"
```

```
    ServerName www.crazylinux.com
```

```
    ErrorLog "logs/www.crazylinux.com.error.log"
```

```
    CustomLog "logs/www.crazylinux.com.access.log" common
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.0.10>
```

```
DocumentRoot "/usr/local/apache2/htdocs/google"
```

```
ServerName www.google.com
```

```
ErrorLog "logs/www.google.com.error.log"
```

```
CustomLog "logs/www.google.com.access.log" common
```

```
</VirtualHost>
```

```
[root@localhost bin]# cd /usr/local/apache2/htdocs/
```

```
[root@localhost bin]# cd /usr/local/apache2/htdocs/
```

```
[root@localhost htdocs]# mkdir crazylinux
```

```
[root@localhost htdocs]# mkdir google
```

```
[root@localhost htdocs]# echo www.crazylinux.com >> crazylinux/index.html
```

```
[root@localhost htdocs]# echo www.google.com >> google/index.html
```

```
[root@localhost htdocs]# service apache restart
```

测试：在 cline 中指定 DNS 服务器的 ip

3) AB 压力测试

先修改最大文件数限制

```
[root@localhost htdocs]# ulimit -a | grep "open files"
```

```
open files                (-n) 1024
```

```
[root@localhost htdocs]# ulimit -n 10000
```

```
[root@localhost htdocs]# ulimit -a | grep "open files"
```

```
open files                (-n) 10000
```

```
[root@localhost htdocs]# /usr/local/apache2/bin/ab -q -c 2000 \
-n 4000 http://www.crazylinux.com/
```

```
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
```

```
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
```

```
Licensed to The Apache Software Foundation, http://www.apache.org/
```

```
Benchmarking www.crazylinux.com (be patient).....done
```


Server Software: Apache/2.2.13
Server Hostname: www.crazylinux.com
Server Port: 80

Document Path: /
Document Length: 19 bytes

Concurrency Level: 2000 //用于并发连接的请求数
Time taken for tests: 8.958 seconds //测试过程消耗的时间
Complete requests: 4000 //测试过程总共完成的请求数
Failed requests: 0 //失败的请求数
Write errors: 0
Total transferred: 1078152 bytes //测试过程网络传输量
HTML transferred: 76152 bytes //测试过程中的 HTML 内容传输量
Requests per second: 446.51 [#/sec] (mean) //服务器平均每秒相应的请求数
Time per request: 4479.196 [ms] (mean) //服务器平均每请求的相应时间
Time per request: 2.240 [ms] (mean, across all concurrent requests)
Transfer rate: 117.53 [Kbytes/sec] received

Connection Times (ms) //连接时间统计信息

	min	mean[+/-sd]	median	max
Connect:	0	535 1262.9	0	4313
Processing:	1	583 1539.2	42	8389
Waiting:	0	583 1539.1	42	8386
Total:	36	1118 2052.6	42	8956

Percentage of the requests served within a certain time (ms)

50%	42
66%	76
75%	864
80%	2489

90% 4321

95% 4609

98% 8872

99% 8909

100% 8956 (longest request)

5、使用 awstats 分析 Web 日志

```
[root@localhost media]# tar -zxvf awstats-6.9.tar.gz -C /usr/local/src/
```

```
[root@localhost media]# mv /usr/local/src/awstats-6.9/ /usr/local/src/awstats
```

```
[root@localhost media]# cd /usr/local/src/awstats/tools/
```

```
[root@localhost tools]# ./awstats_configure.pl
```

And then, run configure.pl from this location.

Do you want to continue setup from this NON standard directory [yN] ? **y**

Warning: You Apache config file contains directives to write 'common' log files

This means that some features can't work (os, browsers and keywords detection).

Do you want me to setup Apache to write 'combined' log files [y/N] ? **y**

Do you want me to build a new AWStats config/profile

file (required if first install) [y/N] ? **y**

-----> Define config file name to create

What is the name of your web site or profile analysis ?

Example: www.mysite.com

Example: demo

Your web site, virtual server or profile name:

> **www.crazylinux.com**

-----> Define config file path

In which directory do you plan to store your config file(s) ?

Default: /etc/awstats

Directory path to store config file(s) (Enter for default):

>

-----> Add update process inside a scheduler

Sorry, configure.pl does not support automatic add to cron yet.

You can do it manually by adding the following command to your cron:

```
/usr/local/src/awstats/wwwroot/cgi-bin/awstats.pl -update -config=www.crazylinux.com
```

Or if you have several config files and prefer having only one command:

```
/usr/local/src/awstats/tools/awstats_updateall.pl now
```

Press ENTER to continue...

Press ENTER to finish...

```
[root@localhost tools]# vim /etc/awstats/awstats.www.crazylinux.com.conf 把 LogFile 修改  
LogFile="/usr/local/apache2/logs/www.crazylinux.com.access.log"
```

```
[root@localhost awstats]# touch /usr/local/apache2/logs/www.crazylinux.com.access.log
```

在此先把 access.log 的日志格式修改一下

这里选择日志的格式为 **combined**,而不是 common, awstats 解析不了 common

```
[root@localhost awstats]# cat /usr/local/apache2/conf/extra/httpd-vhosts.conf
```

```
NameVirtualHost 192.168.0.10
```

```
<VirtualHost 192.168.0.10>
```

```
    DocumentRoot "/usr/local/apache2/htdocs/crazylinux"
```

```
    ServerName www.crazylinux.com
```

```
    ErrorLog "logs/www.crazylinux.com.error.log"
```

```
    CustomLog "logs/www.crazylinux.com.access.log" combined
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.0.10>
```

```
    DocumentRoot "/usr/local/apache2/htdocs/google"
```

```
    ServerName www.google.com
```

```
    ErrorLog "logs/www.google.com.error.log"
```

```
    CustomLog "logs/www.google.com.access.log" common
```

```
</VirtualHost>
```

```
[root@localhost awstats]# perl /usr/local/src/awstats/wwwroot/cgi-bin/awstats.pl -update  
-config=www.crazylinux.com
```

```
Create/Update database for config "/etc/awstats/awstats.www.crazylinux.com.conf" by  
AWStats version 6.9 (build 1.925)
```

```
From data in log file "/usr/local/apache2/logs/www.crazylinux.com.access.log"...
```

```
Phase 1 : First bypass old records, searching new record...
```

Direct access after last parsed record (after line 57)

Jumped lines in file: 57

Found 57 already parsed records.

Parsed lines in file: 8675

Found 0 dropped records,

Found 0 corrupted records,

Found 0 old records,

Found 8675 new qualified records.

```
[root@localhost tmp]# crontab -l
```

```
*/5 * * * /usr/bin/perl /usr/local/src/awstats/wwwroot/cgi-bin/awstats.pl -update  
-config=www.crazylinux.com
```

```
[root@localhost tmp]# service crond restart
```

Stopping crond: [OK]

Starting crond: [OK]

```
[root@localhost tmp]# chkconfig crond on
```

在之前把 APACHE 全局的 CustomLog 注释掉

```
#CustomLog "logs/access_log" combined
```

测试:

```
[root@localhost ~]# cat /usr/local/apache2/htdocs/crazylinux/awstats.html
```

```
<html>
```

```
<head>
```

```
<meta http-equiv=refresh content="0;url=http://www.crazylinux.com/awstats/awstats.pl?  
config=www.crazylinux.com">
```

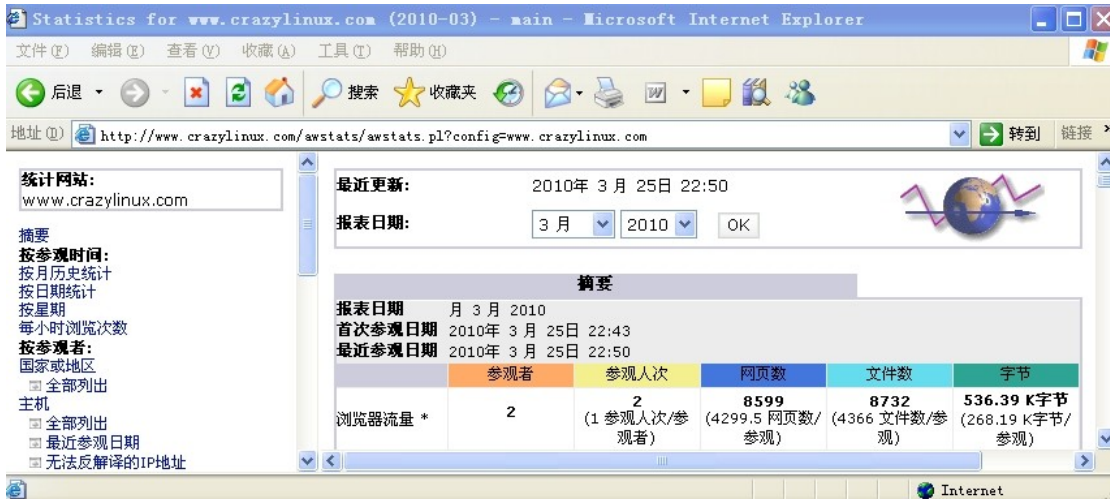
```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

在 client 上 IE 中输入 <http://www.crazylinux.com/awstats.html>



6、对 awstats 进行访问控制

[root@localhost ~]# vim /usr/local/apache2/conf/httpd.conf 在里面查到与 awstats 相关信息

```
<Directory "/usr/local/src/awstats/wwwroot">
    AuthName "Awstats"
    AuthType Basic
    AuthUserFile /usr/local/src/awstats/wwwroot/.htpasswd
    require user awuser
    Options None
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>
```

[root@localhost ~]# /usr/local/apache2/bin/htpasswd

-c /usr/local/src/awstats/wwwroot/.htpasswd awuser

New password:

Re-type new password:

Adding password for user awuser

[root@localhost ~]# service apache restart

测试: 在 client 中 IE 输入 <http://www.crazylinux.com/awstats.html>



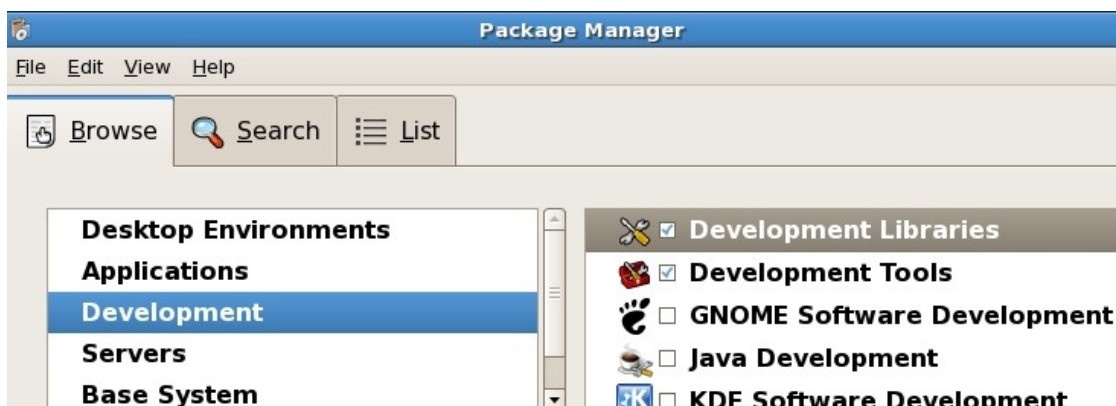
7、安装配置 MYSQL

```
[root@localhost media]# tar -zxvf mysql-5.0.75.tar.gz -C /usr/local/src/
```

```
[root@localhost media]# cd /usr/local/src/mysql-5.0.75/
```

```
[root@localhost mysql-5.0.75]# useradd -M -s /sbin/nologin mysql
```

```
[root@localhost mysql-5.0.75]# yum -y install libtermcap-devel
```



```
[root@localhost mysql-5.0.75]# ./configure --prefix=/usr/local/mysql --with-  
mysqlduser=mysql
```

```
[root@localhost mysql-5.0.75]# make
```

```
[root@localhost mysql-5.0.75]# make install
```

```
[root@localhost mysql-5.0.75]# cp support-files/my-medium.cnf /etc/my.cnf
[root@localhost mysql-5.0.75]# /usr/local/mysql/bin/mysql_install_db --user=mysql
[root@localhost mysql-5.0.75]# chown -R root:mysql /usr/local/mysql/
[root@localhost mysql-5.0.75]# chown -R mysql /usr/local/mysql/var/
[root@localhost mysql-5.0.75]# echo "/usr/local/mysql/lib/mysql" >> /etc/ld.so.conf
[root@localhost mysql-5.0.75]# ldconfig
[root@localhost mysql-5.0.75]# /usr/local/mysql/bin/mysqld_safe --user=mysql &
[root@localhost mysql-5.0.75]# ps -e | grep mysql 验证是否成功开启 mysql
1664 pts/2  00:00:00 mysqld_safe
1690 pts/2  00:00:00 mysqld
[root@localhost mysql-5.0.75]# netstat -ntulp | grep mysql
tcp  0 0 0.0.0.0:3306  0.0.0.0:*      LISTEN    1690/mysqld
[root@localhost mysql-5.0.75]# cp support-files/mysql.server /etc/init.d/mysqld
[root@localhost mysql-5.0.75]# chmod +x /etc/init.d/mysqld
[root@localhost mysql-5.0.75]# chkconfig --add mysqld
[root@localhost mysql-5.0.75]# chkconfig mysqld on
[root@localhost mysql-5.0.75]# export PATH=$PATH:/usr/local/mysql/bin/
[root@localhost mysql-5.0.75]# echo "PATH=$PATH:/usr/local/mysql/bin/" >> /etc/profile
[root@localhost mysql-5.0.75]# mysqladmin -u root password 123456 //修改 mysql 密码
```

8、安装配置 PHP

```
[root@localhost media]# tar -jxvf php-5.2.10.tar.bz2 -C /usr/local/src/
[root@localhost media]# cd /usr/local/src/php-5.2.10/
[root@localhost php-5.2.10]# ./configure --prefix=/usr/local/php5
--enable-mbstring --with-apxs2=/usr/local/apache2/bin/apxs
--with-mysql=/usr/local/mysql --with-config-file-path=/usr/local/php5
[root@localhost php-5.2.10]# make && make install
[root@localhost php-5.2.10]# cp php.ini-dist /usr/local/php5/php.ini
在 APACHE 里添加 php，使 APACHE 支持 PHP
[root@localhost php-5.2.10]# vim /usr/local/apache2/conf/httpd.conf
LoadModule php5_module      modules/libphp5.so
AddType application/x-httpd-php .php
```

```
<IfModule dir_module>
```

```
    DirectoryIndex index.php index.html
```

```
</IfModule>
```

```
[root@localhost php-5.2.10]# cat /usr/local/apache2/htdocs/crazylinux/index.php
```

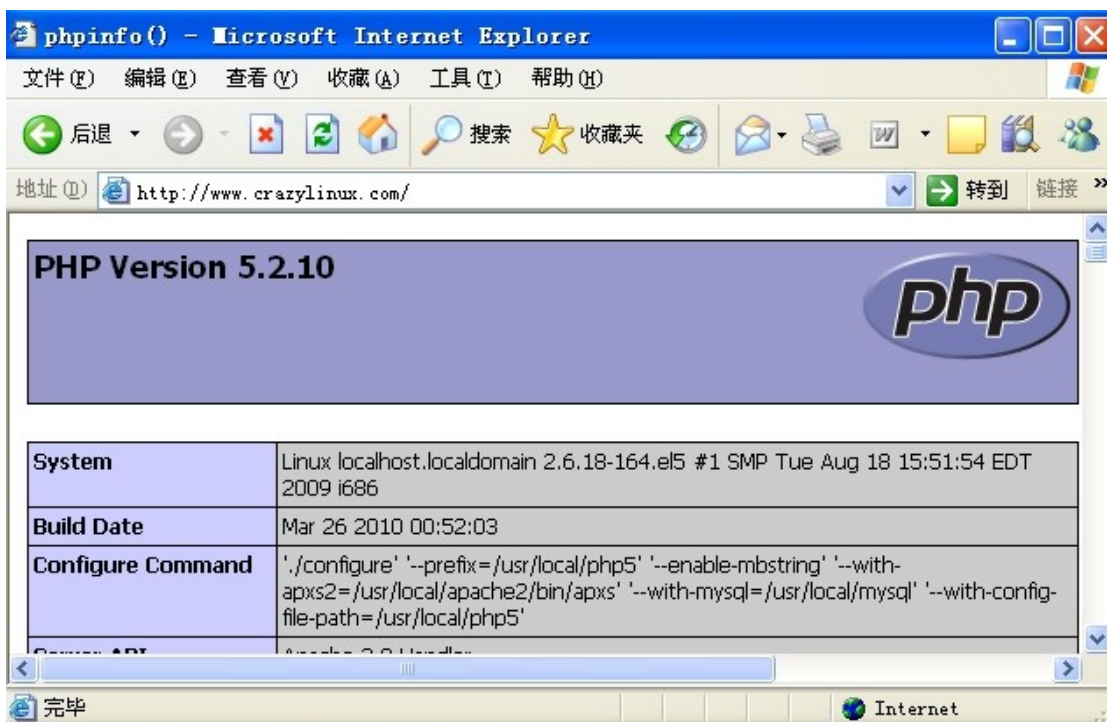
```
<?php
```

```
    phpinfo();
```

```
?>
```

```
[root@localhost php-5.2.10]# service apache restart
```

测试：在 client 的 IE 中输入 <http://www.crazylinux.com>



9、测试 PHP 与 APACHE 的协同工作

```
[root@localhost php-5.2.10]# cd /usr/local/apache2/htdocs/crazylinux/
```

```
[root@localhost crazylinux]# cat testdb.php
```

```
<?php
```

```
    $link=mysql_connect('localhost','root','123456');  
    if(!$link)  
        echo "Fail !!";  
    else  
        echo "Success !!";  
    mysql_close();
```

```
?>
```

在 APACHE 本机测试：（在 client 测试的话需要 mysql 授权，比较麻烦）

需要先安装 firefox 浏览器



10、架设 Discuz 论坛

```
[root@localhost media]# cp Discuz_7.2_FULL_SC_UTF8.zip /home/
```

```
[root@localhost media]# mysql -uroot -p
```

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 63

Server version: 5.0.75-log Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> create database bbsdb;
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> grant all on bbsdb.* to runbbs@localhost identified by '123456';
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> quit
```

Bye

```
[root@localhost media]# cd /home/
```

```
[root@localhost home]# unzip Discuz_7.2_FULL_SC_UTF8.zip -d discuz
```

```
[root@localhost home]# mv discuz/upload/ /usr/local/apache2/htdocs/google/bbs
```

```
[root@localhost bbs]# chown -R daemon config.inc.php attachments forumdata/  
uc_client/data/cache/
```

```
[root@localhost bbs]# mv install/ install.lock
```

```
[root@localhost bbs]# chmod o-rx install.lock/
```

11、安装 phpMyAdmin 数据库管理套件

```
[root@localhost media]# tar -zxvf phpMyAdmin-3.3.1-all-languages.tar.gz
```

```
-C /usr/local/apache2/htdocs/google/
```

```
[root@localhost media]# cd /usr/local/apache2/htdocs/google/
```

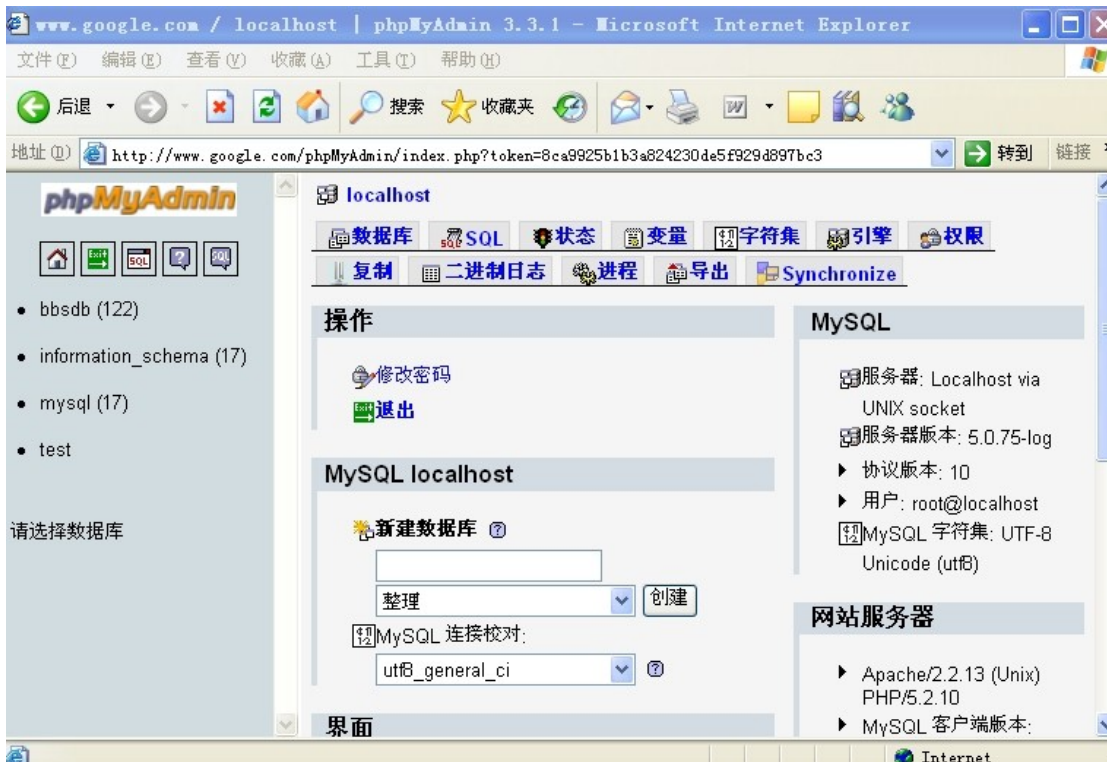
```
[root@localhost google]# mv phpMyAdmin-3.3.1-all-languages/ phpMyAdmin
```

```
[root@localhost google]# cd phpMyAdmin/
```

```
[root@localhost phpMyAdmin]# cp config.sample.inc.php config.inc.php
```

```
[root@localhost phpMyAdmin]# vim config.inc.php
```

```
$cfg['blowfish_secret'] = '123456'; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
```



网友热评

热点技术评论

[unix 下对文件的列值排序比较问题](#)

[sybase 索引数据物理存储结构分析](#)

[sed 可以编辑 excel 文件吗？](#)

[如何做数据迁移，且保持数据的连续性？](#)

[请问大家都用 bsd 做什么？](#)

[AIX 下创建卷组失败](#)

[请教一个正则表达式中 “/e” 的问题](#)

[向 socket 一次写多少数据合适？](#)

[用 perl 做界面有什么不妥之处吗？](#)

[C++ 的高级特性你们是怎么学会的？](#)

[再论 Gtkmm 与 Qt](#)

[请教用 awk 求最大值的问题](#)

[请教 NBU vault 的配置](#)

[在试用 UBUNTU 10.04 beta1](#)

[awk "最靠近" 的算法](#)

[函数重载会降低效率吗](#)

[关于 {} 与 \(\) 的问题](#)

[你使用那个 shell 多一点？](#)

[问一个有关 sed 匹配模式的话题](#)

[虚拟机下的 linux 启动不了](#)

[ip 碎片重组疑问](#)

[如何实现每次开机，linux 自动还原](#)

[关于文法识别算法](#)

[关于 LVS 的粘滞问题](#)

[linux 编译 glibc，一直报错](#)

[yum 安装 samba 报错](#)

[linux 下 su 的疑问](#)

热点新闻评论

[IDC 机房中碰到很奇怪的网络故障](#)

[FreeBSD 搞个桌面挺麻烦啊](#)

[李开复：Vista 系统失败的真正内幕](#)

[提问老 F，CU 有价值观否？](#)

[想知道 NetBSD 主要用途是什么](#)

[现在高校学 linux 的学生是否越来越少了？](#)

[往 linux 方面发展还是坚持学 C++](#)

[如何成长为顶级 linux 系统维护？](#)

[C++ 的 cout 就是个杯具](#)

[如何区分 BSD 系统还是 Linux 系统？](#)

[C++ 是不是在走下坡路呀？](#)

[学 Qt 有没有前途？](#)

[关于 CNNIC 证书的安全隐患](#)

[kernel.org 被黑了](#)

[有感 redhat 和 microsoft 停止支持安腾](#)

[XP 无法正常关机和重启](#)

[看我是如何说服全公司换装 linux 的！](#)

[你觉得 android 有前途不？](#)

[大家有在笔记本上安装 linux 吗？](#)

[黑客声称 Windows 7 比 Linux 安全](#)

[谈谈自己对工作的迷茫](#)

[关于对 linus 等人的不满意](#)

[大家认为不错的 linux 书](#)

[Linux 是怎样败下阵来的？](#)

[模仿 linux 编写的操作系统](#)

[放出 mouseOS 0.01 预览版](#)

[共享 RFC 集合，linux 网络源码伴侣](#)